LINK UTILI

http://www.cplusplus.com/reference

https://www.mrwebmaster.it/programmazione/guida-cplusplus/

http://programmazione.html.it/guide/leggi/34/guida-c/

http://www.bo.cnr.it/corsi-di-informatica/corsoCstandard/Lezioni/01Indice.html#funzioni

http://www.francy59.altervista.org/informatica\_3anno.php

http://www.francy59.altervista.org/informatica\_4anno.php

--------------------------------------------------------------------------

LIBRERIE

//librerie da includere per poterne utilizzare funzionalità aggiuntive

#include <cstdlib> //funzioni della libreria standard del C - \_beep()

#include <iostream> //cin e cout

#include <ctime> //rand - older: time.h

#include <cmath> //funz matematiche

#include <stdio.h> //system("pause"); getchar(); fflush(stdin)

#include <conio.h> //\_getch(); getch() NON VA; kbhit() NON VA sostituisco con \_getch();

#include <windows.h> //Sleep(ms), PlaySound()

#include <string>

#include <sstream>

#include <fstream>

#include <iomanip> //setprecision(3), setw

#include <unistd.h> //usleep(ms); //other Sleep()

#include <pthread.h>

#include <graphics.h> //libreria grafica ereditata dal Borland C++

#include <mmsystem.h> // mciSendString()

#include <vector> //libreria per utilizzare la classe vector

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

TIPI DI DATO: DICHIARAZIONI DI VARIABILI E ASSEGNAZIONI

definizioni variabili: tipo nome;

int x; //x intera

float y; //y reale

double z; //z reale a doppia precisione

char c; //c un singolo carattere

string s; //s sequenza di caratteri/cifre

assegnazioni

int x=10;

float y=3.4;

char p='A';

string n="paolo";

Assegnazioni (il punto decimale anglosassone in C++ corrisponde alla nostra virgola)

x=3;

y=3.58;

z=7867812424.24242

c='W';

s="daniele";

Assegnazioni utili

conta=conta+1;

conta=conta-1;

somma=somma+numero;

//input

cin>>nome\_variabile;

//output var

cout<<nome\_variabile

//output messaggio testuale

cout<<"TESTO VISUALIZZAO A VIDEO";

//output mix (messaggio e var)

cout<<"Hai fatto "<<n"<<" punti";

//input stringhe con spazi

string frase;

fflush(stdin); //ripulire la memoria della tastiera (prima di una cin/getline/getchar)

getline(cin, frase);

cout<<frase;

//pausa finchè non si preme INVIO

getchar();

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ESCAPE SEQUENCE

cout<<"\t";

cout<<"\n";

cout<<endl;

--------------------------------------------------------------------------

CARATTERI SPECIALI

//quadre

[ aperta quadra: alt+91 (tast. numerico)

] chiusa quadra: alt+93 (tast. numerico)

//graffe

{ aperta graffa: alt+123 (tast. numerico)

} chiusa graffa: alt+125 (tast. numerico)

//€ euro: ALT+0128

//visualizza il simbolo dell'euro

void print\_euro(){

SetConsoleOutputCP(1252);

cout<<(char)(128);

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

CONDIZIONI (DA METTERE NELLA SELEZIONE if O NELLE CONDIZIONI DEI CICLI ITERATIVI)

x==3

x==y

x>y

x>=y

x<y

x<=y

x!=y

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

SELEZIONE

if(condizione1){

/\*blocco di istruzioni

eseguito quando la

condizione1 è vera

\*/

}

else if(condizione1){

/\*blocco di istruzioni

eseguito quando la

condizione2 è vera

\*/

}

else{

/\*blocco di istruzioni

eseguito quando le

precedenti

condizioni non sono verae

\*/

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

SELEZIONE MULTIPLA CON LO SWITCH

int scelta;

cin>>scelta;

switch (scelta){

case 1:

// istruzioni

break;

case 2:

// istruzioni

break;

//altri casi

case 10:

// istruzioni

break;

default:

// istruzioni

break;

}

NB Utile per creare i menu di scelta

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ITERAZIONI CON CONTATORE

int conta;

for(conta=1; conta<=10; conta++){

/\*blocco di istruzioni

eseguito 10 volte

\*/

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ITERAZIONI CON CONDIZIONE INZIALE

int conta=1;

while(conta<=10){

/\*blocco di istruzioni

eseguito 10 volte

\*/

conta++;

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ITERAZIONI CON CONDIZIONE FINALE

int conta=1;

do{

/\*blocco di istruzioni

eseguito 10 volte

\*/

conta++;

}while(conta<=10);

--------------------------------------------------------------------------

ATTESA TIME SLEEP

//attesa per x ms

Sleep(x); //pausa per x ms

//ES

for(int i=0; i<30; i++){

cout<<".";

Sleep(100); //pausa per 0.1 sec

}

--------------------------------------------------------------------------

NUMERI CASUALI

srand((unsigned)time(NULL)); //seme (come mischiare il mazzo di carte)

int MAX=6;

int MIN=1;

int numero;

numero=rand()%(MAX-MIN+1)+MIN; //genero un numero intero casuale compreso fra 1 e 6

int MAX=30;

int MIN=18;

int numero;

numero=rand()%(MAX-MIN+1)+MIN; //genero un numero intero casuale compreso fra 18 e 30

FUNZIONE CASUALE

#include <cstdlib> //funzioni della libreria standard del C

#include <iostream> //cin e cout

#include <ctime> //rand

#include <cmath> //funz matematiche

#include <stdio.h> //getchar()

#include <windows.h> //Sleep(), PlaySound()

#include <iomanip> // std::setw

#include <time.h>

#include <fstream>

#include <string>

#include <sstream>

using namespace std;

//inizializza generatore di numeri pseudo-casuali

void seme(){

srand((unsigned)time(NULL)); //seme (come mischiare il mazzo di carte)

}

//genera un numero intero casuale compreso fra MIN e MAX

int casuale(int MIN, int MAX){

return(rand()%(MAX-MIN+1)+MIN);

}

//Genero MAX-MIN+1 numeri interi casuali senza ripetizioni compresi fra MIN e MAX

//li inserisco in un vettore la cui dimensione è MAX-MIN+1

void randVet(int vet[], int MIN, int MAX){

seme();

int i=0, j;

bool trovato;

vet[i]=casuale(MIN, MAX);

for(i=1; i<MAX-MIN+1; i++){

do{

vet[i]=casuale(MIN, MAX);

j=0;

trovato=false;

do{

if(vet[i]==vet[j])

trovato=true;

j++;

}while(trovato==false && j<i);

}while(trovato==true);

}

}

int main(int argc, char \*argv[]){

int MIN=18;

int MAX=30;

int LEN=MAX-MIN+1; //lunghezza del vettore

int vet[LEN];

randVet(vet, MIN, MAX);

cout<<"Casuali\n";

for(int i=0; i<LEN; i++)

cout<<vet[i]<<endl;

MIN=2;

MAX=10;

srand((unsigned)time(NULL)); //mischio

cout<<"\nAltri\n";

for(int i=0; i<60; i++)

cout<<casuale(MIN, MAX)<<endl; //genero un numero compreso fra MIN e MAX

system("pause");

return 0;

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

CONVERSIONI

string charTOstring(char s[]){

string str;

str.assign(s);

return(str);

}

char \*stringTOchar(string s){

int len=s.length();

char \*ss=new char[len+1];

s.copy(ss,len,0);

ss[len]='\0';

return(ss);

}

string intTOstring(int value){

stringstream ss;

ss<<value;

return(ss.str());

}

char\* intTOchar(int value){

return(stringTOchar(intTOstring(value)));

}

const char\* stringTOconstChar(string s){

const char \*ss=s.c\_str();

return(ss);

}

int stringTOint(string s){

return(atoi(stringTOconstChar(s)));

}

bool is\_numeric(string str){

int len=str.length();

int number=0;

for(int i=0; i<len; i++){ ;

// check for decimal digits

if(isdigit(str[i])!=0)

number++;

}

if(number==len)

return true;

else

return false;

}

--------------------------------------------------------------------------

COMANDI DI SISTEMA

//richiamare il prompt dei comandi MS-DOS

You can execute Windows Command prompt commands using a C++ function called system();

#include <iostream>

system("CMD\_COMMAND");

system("TITLE titolo\_Finestra"); //titolo della finestra di esecuzione del programma

system("CLS"); //ripulire lo schermo

system("PAUSE");

system("dir");

system("calc");

system("DATE");

system("notepad"); //To open a text file

system("start excel"); //To open an excel file

system("start winword"); //To open a word doc file

system("start 7zFM"); //To open a 7z file

system("start devenv"); //To open a visual studio file

To open a file file name of "myfile.txt" with "notepad.exe"

system( "notepad.exe myfile.txt" );

s = "notepad.exe myfile.txt";

system(s.c\_str());

--------------------------------------------------------------------------

//colori testo e sfondo

Come prima cosa devi includere le librerie windows.h e conio.h

La stringa per cambiare i colori è: system("color XY");

Dove X è il colore per lo sfondo e Y per il testo

I colori disponibili sono:

0 = nero

1 = blu scuro

2 = verde

3 = verde acqua

4 = bordeaux

5 = viola

6 = verde oliva

7 = grigio chiaro

8 = grigio

9 = blu

A = verde limone

B = azzurro

C = rosso

D = fucsia

E = giallo

F = bianco

Es: system("COLOR 2c"); //sfondo verde (2) e colore testo rosso (c)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

//sposta il cursore alle coordinate row, col

void gotoXY(int row, int col){

COORD coor;

HANDLE hOutput;

hOutput=GetStdHandle(STD\_OUTPUT\_HANDLE);

coor.X=col;

coor.Y=row;

SetConsoleCursorPosition(hOutput,coor);

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Eseguo l'input su una variabile del tipo string e verifico se è numerico o non

bool is\_numeric(string str){

int len=str.length();

int number=0;

for(int i=0; i<len; i++){ ;

// check for decimal digits

if(isdigit(str[i])!=0)

number++;

}

if(number==len)

return true;

else

return false;

}

/\*

void what\_is(string str){

int len=str.length();

int alphabet=0, number=0;

for(int i=0; i<len; i++){

// check for alphabets

if(isalpha(str[i])!=0)

alphabet++;

// check for decimal digits

else if(isdigit(str[i])!=0)

number++;

}

printf("Alphabetic\_letters = %d, Decimal\_digits = %d\n", alphabet, number);

}

\*/

--------------------------------------------------------------------------

//funzioni matematiche

y=abs(x); //valore assoluto intero

y=fabs(x); //valore assoluto float

y=ceil(x); //arrotondamento per eccesso

y=floor(x); //arrotondamento per difetto

p=pow(b,e); //potenza dove b=base ed e=esponente

p=pow10(e); //potenza del 10

y=sqrt(x); //radice quadrata

M\_PI; // pi greco

y=sin/cos/tan(x); //seno/coseno/tangente di x

y=exp(x); //numero di Nepero

y=log10(x); //logaritmo in base 10 di x

ES:

float x=3;

float y=sqrt(x); cout<<y;

float z=pow(x, 3); cout<<z;

int w=-3; cout<<abs(w);

int a=3, b=5; swap(a, b);cout<<a<<"-"<<b;

cout<<min(x, y); cout<<max(x, y);

cout<<ceil(3.42); cout<<floor(5.16);

cout<<sin(0.8); cout<<cos(0.5);

seme(); cout<<casuale(1, 10);

cout<<RadiantToDegree(M\_PI)<<endl;

cout<<sin(M\_PI/2)<<endl;

--------------------------------------------------------------------------

FUNZIONI GESTIONE STRINGHE

ricerca stringhe di testo

string s;

cout<<"Stringa di testo: ";

getline(cin, s); //al posto di cin ... riconosce gli spazi

string scerca;

cout<<"Stringa da cercare: ";

getline(cin, scerca);

int pos=s.find(scerca, 0);

if(pos!=-1)

cout<<"Stringa trovata\n";

else

cout<<"Stringa non trovata\n";

cout<<pos<<endl;

LUNGHEZZA

string msg="benvenuto";

int lunghezza-msg=msg.length();

CONCATENAZIONE

string saluto="ciao", chi="mondo", message=saluto + ", " + chi + "!";

CONFRONTO ==, !=, >, <, >=, <=

string s1, s2;

if(s1==s2)

cout<<"Stringhe uguai"<<endl;

ACCESSO AGLI ELEMENTI DELLA STRINGA

string s="daniele"; (il primo carattere ha posizione 0 nella stringa)

char c;

c=s.at(2); //c=s[2]; c<--'n'

INSERIMENTO

string s="Vittorio Secondo", s1="Emanuele";

//start-pos=8 ... a partire dalla posizione 8 inserisci s1 in s

s.insert(8, s1); s<--"Vittorio Emanuale Secondo"

ESTRAZIONE SOTTOSTRINGA

string s="Vittorio Secondo";

string s1=s.substr(8, 7); //a partire dalla pos 8 estrai 7 char s1<--"Secondo"

CANCELLAZIONE SOTTOSTRINGA

string s="Vittorio Secondo";

s.erase(8, 7); ////a partire dalla pos 8 cancella 7 char s1<--"Secondo"

SOSTITUZIONE SOTTOSTRINGA

string s="Vittorio Secondo";

s.replace(0, 8, "Umberto"); //dalla pos 0 sostituisci 8 char di s con "Umberto"

s<--"Umberto Secondo"

RICERCA SOTTOSTRINGA

string s="Vittorio Secondo";

int pos=s.find("S", 0); //a partire dalla pos 0 cerca S

pos<--9

DIRECTORY CORRENTE

string ExePath(){

char buffer[MAX\_PATH];

GetModuleFileName(NULL, buffer, MAX\_PATH);

string::size\_type pos=string(buffer).find\_last\_of("\\/");

return string(buffer).substr(0, pos);

}

const char\* stringTOconstChar(string s){

const char \*ss=s.c\_str();

return(ss);

}

int main(int argc, char \*argv[]){

char percorso[MAX\_PATH];

strcpy(percorso, stringTOconstChar(ExePath()));

cout<<"my directory is "<<percorso<<"\n"; //directory corrente

system("PAUSE>null");

return 0;

}

DIRECTORY E FILE DELLA DIRECORY CORRENTE

void search(char \*p){

WIN32\_FIND\_DATA fd;

char pf[MAX\_PATH];

strcpy(pf, p);

strcat(pf, "\\\*.\*");

HANDLE h=FindFirstFile(pf, &fd);

if(h!=INVALID\_HANDLE\_VALUE){

do{

if(strcmp(fd.cFileName, ".") && strcmp(fd.cFileName, "..")){

if(fd.dwFileAttributes & FILE\_ATTRIBUTE\_DIRECTORY){

printf("[%s]\n", fd.cFileName);

char pf[MAX\_PATH];

strcpy(pf, p);

strcat(pf, "\\");

strcat(pf, fd.cFileName);

search(pf);

}

else

printf("\t%s\n", fd.cFileName);

}

}while(FindNextFile(h, &fd));

FindClose(h);

}

else

printf("Errore %d\n", GetLastError());

}

int main(int argc, char \*argv[]){

cout<<"my directory is "<<ExePath()<<"\n"; //directory corrente

char percorso[MAX\_PATH];

strcpy(percorso, stringTOconstChar(ExePath()));

//cout<<"Inserire il percorso in effettuare la ricerca: ";

///cin>>percorso;

//strcpy(percorso, "i:\\Android");

//percorso="i:\\Android";

search(percorso); //es: i:\\Android

system("PAUSE>null");

return 0;

}

SOSTITUISCI TUTTO sostituisce tutte le occorrenze della stessa substring in una string

string ReplaceAll(string str, const string& from, const string& to){

size\_t start\_pos=0;

while((start\_pos=str.find(from, start\_pos))!=string::npos){

str.replace(start\_pos, from.length(), to);

start\_pos+=to.length(); // Handles case where 'to' is a substring of 'from'

}

return str;

}

string ExePath(){

char buffer[MAX\_PATH];

GetModuleFileName(NULL, buffer, MAX\_PATH);

string::size\_type pos=string(buffer).find\_last\_of("\\/");

return string(buffer).substr(0, pos);

}

const char\* stringTOconstChar(string s){

const char \*ss=s.c\_str();

return(ss);

}

int main(int argc, char \*argv[]){

//sostituzione totale

string str\_partenza, str\_arrivo, sub\_str\_old, sub\_str\_nuovo;

str\_partenza="Number Of Beans";

sub\_str\_old=" ";

sub\_str\_nuovo="\_";

str\_arrivo=ReplaceAll(string(str\_partenza), string(sub\_str\_old), string(sub\_str\_nuovo));

cout<<str\_arrivo<<endl;

str\_partenza="Number Of Beans";

sub\_str\_old=" "; //due spazi

sub\_str\_nuovo="\_";

str\_arrivo=ReplaceAll(string(str\_partenza), string(sub\_str\_old), string(sub\_str\_nuovo));

cout<<str\_arrivo<<endl;

str\_partenza="Number Of Beans";

sub\_str\_old="x";

sub\_str\_nuovo="\_"; //nessuna occorrenza

str\_arrivo=ReplaceAll(string(str\_partenza), string(sub\_str\_old), string(sub\_str\_nuovo));

cout<<str\_arrivo<<endl;

//fine

system("PAUSE>null");

return 0;

}

SOSTITUZIONI OCCORRENZE FRA <TAG> E </TAG>

string StrReplaceBetweenTag(string Partenza, string NuovaStringa, string Tag){

// Inizializzo le sottostringhe iniziali e finali del Tag

string StartTag="<"+Tag+">";

string EndTag="</"+Tag+">";

// Posizione del tag nella string di partenza

int Start=Partenza.find(StartTag);

// Se il tag è presente nella stringa continuo

// Altrimenti ritorno la stringa di partenza

if (Start>-1){

//Posiziono l'inizio della sostituzione, dopo il tag di apertura

Start+=StartTag.size();

// Posizione del tag di chiusura

int End=Partenza.find(EndTag);

if(End>-1){

int Length=End-Start;

Partenza.replace(Start, Length, NuovaStringa);

}

else

Partenza.replace(Start, Partenza.size(), NuovaStringa);

return Partenza;

}

else

return Partenza;

}

int main(int argc, char \*argv[]){

string Partenza="eefrf erfekjrfjernfkenkrjfnejrnfek <str>edjwnekjfnwkefnw</str> blabla";

string NewString="prova";

string Tag="str";

cout<<StrReplaceBetweenTag(Partenza, NewString, Tag)<<endl;

system("PAUSE>null");

return 0;

}

--------------------------------------------------------------------------

GRAFICA

#include <windows.h>

//spostare il cursore a determinate coordinate

/\*

(0,0)----------------------->row

|

|

|

|

|

|

col

\*/

void gotoXY(int row, int col){

COORD coor;

HANDLE hOutput;

hOutput=GetStdHandle(STD\_OUTPUT\_HANDLE);

coor.X=col;

coor.Y=row;

SetConsoleCursorPosition(hOutput, coor);

}

int row=16, col=16;

gotoXY(row, col);

--------------------------------------------------------------------------

ASCII

char c='a';

cout<<"Il codice ASCII per "<<c<<" e' "<<(int)c;

--------------------------------------------------------------------------

//data

struct Tdata{

int anno;

int mese;

int giorno;

int ora;

int min;

int sec;

};

Tdata getData(){

Tdata dataOggi;

time\_t data;

tm \*dataInfo;

data=time(NULL); // ottiene l'ora odierna

dataInfo=localtime(&data);

dataOggi.anno=dataInfo->tm\_year+1900;

dataOggi.mese=1+dataInfo->tm\_mon;

dataOggi.giorno=dataInfo->tm\_mday;

dataOggi.ora=dataInfo->tm\_hour;

dataOggi.min=dataInfo->tm\_min;

dataOggi.sec=dataInfo->tm\_sec;

return(dataOggi);

}

Tdata dataOggi;

dataOggi=getData();

cout<<"A: "<<dataOggi.anno;

--------------------------------------------------------------------------

/\*

ceil() x eccesso

floor() x difetto

\*/

//approssima il numero num a d decimali

double round(double num, int dec){

return((floor(num\*pow(10, dec)))/pow(10, dec));

}

--------------------------------------------------------------------------

Dev-Cpp\_ModByCorti noG

#include <iostream>

#include <windows.h>

#include <conio.h>

using namespace std;

int main(){

bool repeat=true;

while(repeat){

cout<<'\*';

Sleep(100);

if(kbhit()){

char ch=getch();

switch(ch){

case 27: // press ESC to exit

repeat = false;

break;

case 32: // press SPACE to clear screen

clrscr();

break;

}

}

}

return 0;

}

--------------------------------------------------------------------------

graphics.h color

Constante Valor Significado

BLACK 0 Negro

BLUE 1 Azul

GREEN 2 Verde

CYAN 3 Cían

RED 4 Rojo

MAGENTA 5 Magenta

BROWN 6 Marrón

LIGHTGRAY 7 Gris Claro

DARKGRAY 8 Gris Oscuro

LIGHTBLUE 9 Azul Claro

LIGHTGREEN 10 Verde Claro

LIGHTCYAN 11 Cían Claro

LIGHTRED 12 Rojo Claro

LIGHTMAGENTA 13 Magenta Claro

YELLOW 14 Amarillo

WHITE 15 Blanco

Description:

This page provides a mechinism for using the WinBGIm Graphics Library with the Dev-C++ development environment.. Additional documentation for the WinBGIm graphics library is available at www.cs.colorado.edu/~main/bgi.

Installation Notes:

Install Dev-C++. I installed from the Version 4.9.9.2 Setup File.

Download graphics.h to the include/ subdirectory of the Dev-C++ directories.

Download libbgi.a to the lib/ In order to use the WinBGIm subdirectory of the Dev-C++ directories.

Whenever you #include <graphics.h> in a program, you must instruct the linker to link in certain libraries. The command to do so from Dev-C++ is Alt-P. Choose the Parameters tab from the pop-up window and type the following into the Linker area:

-lbgi

-lgdi32

-lcomdlg32

-luuid

-loleaut32

-lole32

You can now compile and run programs that use the WinBGIm graphics library, such as this one that opens a small window, draws a circle and waits for the user to press a key:

#include <graphics.h>

int main( )

{

initwindow(400, 300, "First Sample");

circle(100, 50, 40);

while (!kbhit( ))

{

delay(200);

}

return 0;

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

#define PI 3.14159265

//funzione test per la gestione delle funzioni trigonometriche

void trigo(){

//funz trigonometriche

double x1, x2, y, a, r, p;

x1=-10.0; //cateto 1

x2=10.0; //cateto 2

p=30.0;

r=sin(p\*PI/180);

printf ("The sine of %f degrees is %f.\n", p, r);

x1=4.0;

y=5.0;

//a=asin(x1/y)\*180.0/PI;

cout<<a;

//cout<<atan2(x1, x2)\*180/PI; //arctg in gradi

//result=sin(a\*PI/180); //a: angolo in gradi

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

//funzione test per studio dei cicli iterativi

void test(){

int i, t, t\_min=1000;

for(i=1; i<=10; i++){

cout<<"Tempo sul giro in sec: ";

cin>>t;

if(t<t\_min){

t\_min=t;

}

}

cout<<"Miglior tempo. "<<t\_min<<endl;

t\_min=1000;

i=1;

do{

cout<<"Tempo sul giro in sec: ";

fflush(stdin);

cin>>t;

if(t<t\_min){

t\_min=t;

}

i++;

}while(t<100);

cout<<"Miglior tempo. "<<t\_min<<endl;

int scelta;

fflush(stdin);

cin>>scelta;

switch(scelta){

case 1:

cout<<"Addizione";

break;

case 2:

cout<<"Sottrazione";

break;

case 3:

cout<<"Moltiplicazione";

break;

case 4:

cout<<"Divisione";

break;

default:

cout<<"Scelta errata";

break;

}

int giri=1, num\_giri=10;

int t\_totale=0, t\_medio;

do{

cout<<"Tempo sul giro in sec: ";

fflush(stdin);

cin>>t;

t\_totale=t\_totale+t;

if(t<t\_min){

t\_min=t;

}

giri++;

}while(giri<=num\_giri);

t\_medio=t\_totale/num\_giri;

giri=1;

t\_totale=0;

cin>>num\_giri;

do{

cout<<"Tempo sul giro in sec: ";

fflush(stdin);

cin>>t;

t\_totale=t\_totale+t;

if(t<t\_min){

t\_min=t;

}

giri++;

}while(giri<=num\_giri);

t\_medio=t\_totale/num\_giri;

giri=1;

t\_totale=0;

cin>>num\_giri;

int t\_limite=100;

do{

cout<<"Tempo sul giro in sec: ";

fflush(stdin);

cin>>t;

t\_totale=t\_totale+t;

giri++;

}while(giri<=num\_giri && t<t\_limite);

giri=giri-1;

t\_medio=t\_totale/giri;

for(giri=1; giri<=num\_giri && t<t\_limite; giri++){

fflush(stdin);

cin>>t;

t\_totale=t\_totale+t;

}giri=giri-1;

t\_medio=t\_totale/giri;

double voto, somma, media;

int num\_voti;

fflush(stdin);

cin>>num\_voti;

for(int i=0; i<num\_voti; i++){

fflush(stdin);

cin>>voto;

somma=somma+voto;

}

media=somma/num\_voti;

somma=0;

for(int i=1; i<=10; i++){

fflush(stdin);

cin>>voto;

somma+=voto;

}

media=somma/num\_voti;

int p, x=3;

p+=x; //p=p+x;

p-=x; //p=p-x;

p\*=x; //p=p\*x

p/=x; //p=p/x;

for(i=1; i<=10; i++){

Sleep(500);

cout<<"Ciao\n";

}

i=0;

do{

Sleep(500);

cout<<"Ciao\n";

i++;

}while(i<=10);

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

CONVERSIONI DI TIPO E SPLIT

int stringTOint(string s){

return(atoi(s.c\_str()));

}

float stringTOfloat(string s){

return(atof(s.c\_str()));

}

string intTOstring(int value){

stringstream ss;

ss<<value;

return(ss.str());

}

string floatTOstring(float value){

stringstream ss;

ss<<value;

return(ss.str());

}

void cins(char s[], int dim){

//while(cin.get()!='\n');

//cin.get(s,dim,'\n');

gets(s);

}

string charTOstring(char s[]){

string str;

str.assign(s);

return(str);

}

char \*stringTOchar(string s){

int len=s.length();

char \*ss=new char[len+1];

s.copy(ss,len,0);

ss[len]='\0';

return(ss);

}

char\* intTOchar(int value){

return(stringTOchar(intTOstring(value)));

}

const char\* stringTOconstChar(string s){

const char \*ss=s.c\_str();

return(ss);

}

/\*

const char\* intTOconstChar(int value)

{

return(stringTOconstChar(intTOstring(value)));

}

\*/

void couts(char s[]){

cout<<charTOstring(s);

}

string \*split(string s, int &numwords){

cout<<s<<endl;

char separatore=' ';

int len=s.length();

int sublen=0;

numwords=1;

for(int i=0; i<len; i++)

{

if(s[i]==separatore)

numwords++;

}

//cout<<numwords<<endl;

//string words[numwords];

string \*words=new string[numwords];

words=new string[numwords];

int j=0;

for(int i=0; j<numwords; i++)

{

if(s[i]==separatore)

{

words[j]=s.substr(i-sublen,sublen);

sublen=0;

j++;

}

else

sublen++;

}

//for(int j=0; j<numwords; j++)

//cout<<words[j]<<"\t"<<words[j].length()<<endl;

return(words);

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

get\_key

bool key\_pressed(int key){

return(GetAsyncKeyState(key) & 0x8000!=0);

}

string get\_key(){

//https://docs.microsoft.com/en-us/windows/desktop/inputdev/virtual-key-codes

string key="";

if(key\_pressed(VK\_LEFT)){

key="LEFT";

cout<<key<<endl;

}

else if(key\_pressed(VK\_RIGHT)){

key="RIGHT";

cout<<key<<endl;

}

else if(key\_pressed(VK\_UP)){

key="UP";

cout<<key<<endl;

}

else if(key\_pressed(VK\_DOWN)){

key="DOWN";

cout<<key<<endl;

}

else if(key\_pressed(VK\_F1)){

key="F1";

cout<<key<<endl;

}

else if(key\_pressed(VK\_F2)){

key="F2";

cout<<key<<endl;

}

else if(key\_pressed(VK\_F3)){

key="F3";

cout<<key<<endl;

}

else if(key\_pressed(VK\_F4)){

key="F4";

cout<<key<<endl;

}

else if(key\_pressed(VK\_F5)){

key="F5";

cout<<key<<endl;

}

else if(key\_pressed(VK\_F6)){

key="F6";

cout<<key<<endl;

}

else if(key\_pressed(VK\_F7)){

key="F7";

cout<<key<<endl;

}

else if(key\_pressed(VK\_F8)){

key="F8";

cout<<key<<endl;

}

else if(key\_pressed(VK\_F9)){

key="F9";

cout<<key<<endl;

}

else if(key\_pressed(VK\_F10)){

key="F10";

cout<<key<<endl;

}

else if(key\_pressed(VK\_F11)){

key="F11";

cout<<key<<endl;

}

else if(key\_pressed(VK\_F12)){

key="F12";

cout<<key<<endl;

}

else if(key\_pressed(VK\_TAB)){

key="TAB";

cout<<key<<endl;

}

else if(key\_pressed(VK\_SHIFT)){

key="SHIFT";

cout<<key<<endl;

}

else if(key\_pressed(VK\_CONTROL)){

key="CTRL";

cout<<key<<endl;

}

else if(key\_pressed(VK\_END)){

key="END";

cout<<key<<endl;

}

else if(key\_pressed(VK\_HOME)){

key="HOME";

cout<<key<<endl;

}

else if(key\_pressed(VK\_CANCEL)){

key="CONTROLL-BREAK PROCESSING";

cout<<key<<endl;

}

else if(key\_pressed(VK\_LBUTTON)){

//Left mouse button

key="LBUTTON";

cout<<key<<endl;

}

else if(key\_pressed(VK\_RBUTTON)){

//Right mouse button

key="RBUTTON";

cout<<key<<endl;

}

else if(key\_pressed(VK\_MBUTTON)){

//Middle mouse button

key="MBUTTON";

cout<<key<<endl;

}

else if(key\_pressed(VK\_INSERT)){

key="INS";

cout<<key<<endl;

}

else if(key\_pressed(VK\_DELETE)){

key="DEL";

cout<<key<<endl;

}

else if(key\_pressed(VK\_RETURN)){

key="INVIO";

cout<<key<<endl;

}

else if(key\_pressed(VK\_BACK)){

//backspace key

key="BACK";

cout<<key<<endl;

}

else if(key\_pressed(VK\_MENU)){

key="ALT";

cout<<key<<endl;

}

else if(key\_pressed(VK\_PAUSE)){

key="PAUSE";

cout<<key<<endl;

}

else if(key\_pressed(VK\_CAPITAL)){

key="CAPSLOCK";

cout<<key<<endl;

}

else if(key\_pressed(VK\_SPACE)){

key="SPACEBAR";

cout<<key<<endl;

}

else if(key\_pressed(VK\_PRIOR)){

key="PAGEUP";

cout<<key<<endl;

}

else if(key\_pressed(VK\_NEXT)){

key="PAGEDOWN";

cout<<key<<endl;

}

else if(key\_pressed(0x30)){

key="0";

cout<<key<<endl;

}

else if(key\_pressed(0x31)){

key="1";

cout<<key<<endl;

}

else if(key\_pressed(0x32)){

key="2";

cout<<key<<endl;

}

else if(key\_pressed(0x33)){

key="3";

cout<<key<<endl;

}

else if(key\_pressed(0x34)){

key="4";

cout<<key<<endl;

}

else if(key\_pressed(0x35)){

key="5";

cout<<key<<endl;

}

else if(key\_pressed(0x36)){

key="6";

cout<<key<<endl;

}

else if(key\_pressed(0x37)){

key="7";

cout<<key<<endl;

}

else if(key\_pressed(0x38)){

key="8";

cout<<key<<endl;

}

else if(key\_pressed(0x39)){

key="9";

cout<<key<<endl;

}

else if(key\_pressed(0x41)){

key="A";

cout<<key<<endl;

}

else if(key\_pressed(0x42)){

key="B";

cout<<key<<endl;

}

else if(key\_pressed(0x43)){

key="C";

cout<<key<<endl;

}

else if(key\_pressed(0x45)){

key="E";

cout<<key<<endl;

}

else if(key\_pressed(0x46)){

key="F";

cout<<key<<endl;

}

else if(key\_pressed(0x47)){

key="G";

cout<<key<<endl;

}

else if(key\_pressed(0x48)){

key="H";

cout<<key<<endl;

}

else if(key\_pressed(0x49)){

key="I";

cout<<key<<endl;

}

else if(key\_pressed(0x4A)){

key="J";

cout<<key<<endl;

}

else if(key\_pressed(0x4B)){

key="K";

cout<<key<<endl;

}

else if(key\_pressed(0x4C)){

key="L";

cout<<key<<endl;

}

else if(key\_pressed(0x4D)){

key="M";

cout<<key<<endl;

}

else if(key\_pressed(0x4E)){

key="N";

cout<<key<<endl;

}

else if(key\_pressed(0x4F)){

key="O";

cout<<key<<endl;

}

else if(key\_pressed(0x50)){

key="P";

cout<<key<<endl;

}

else if(key\_pressed(0x51)){

key="Q";

cout<<key<<endl;

}

else if(key\_pressed(0x52)){

key="R";

cout<<key<<endl;

}

else if(key\_pressed(0x53)){

key="S";

cout<<key<<endl;

}

else if(key\_pressed(0x54)){

key="T";

cout<<key<<endl;

}

else if(key\_pressed(0x55)){

key="U";

cout<<key<<endl;

}

else if(key\_pressed(0x56)){

key="V";

cout<<key<<endl;

}

else if(key\_pressed(0x57)){

key="W";

cout<<key<<endl;

}

else if(key\_pressed(0x58)){

key="X";

cout<<key<<endl;

}

else if(key\_pressed(0x59)){

key="Y";

cout<<key<<endl;

}

else if(key\_pressed(0x5A)){

key="Z";

cout<<key<<endl;

}

else if(key\_pressed(VK\_LWIN)){

key="LWIN";

cout<<key<<endl;

}

else if(key\_pressed(VK\_RWIN)){

key="RWIN";

cout<<key<<endl;

}

else if(key\_pressed(VK\_ESCAPE)){

key="ESC";

cout<<key<<endl;

}

//else

return(key);

}

while(!key\_pressed(VK\_ESCAPE)){

if(key\_pressed(VK\_LEFT)){

key="LEFT";

//move(key);

}

else if(key\_pressed(VK\_RIGHT)){

key="RIGHT";

//move(key);

}

else if(key\_pressed(VK\_UP)){

key="UP";

//move(key);

}

else if(key\_pressed(VK\_DOWN)){

key="DOWN";

//move(key);

}

else

is\_key\_correct\_pressed=false;

if(is\_key\_correct\_pressed==true){

move(key);

aggiorna\_punti();

mostra\_punteggio();

mostra\_oggetti();

}

else

is\_key\_correct\_pressed=true;

}

/\*

int c;

while(true)

{

//if(GetAsyncKeyState(VK\_F12))

//if(GetAsyncKeyState(VK\_LBUTTON))

if(GetAsyncKeyState(VK\_BACK))

{

c++;

cout<<c;

}

Sleep(10); // NEW LINE OF CODE - pauses for 10 milliseconds

}

\*/

/\*

const char ESC=27;

const char ENTER=13;

const char UP=72;

const char DOWN=80;

const char SX=75;

const char DX=77;

const char F1=59;

const char F2=60;

const char F3=61;

const char F4=62;

const char F5=63;

const char F6=64;

const char F7=65;

const char F8=66;

const char F9=67;

const char F10=68;

const char F11=133;

const char F12=134;

const char INS=82;

const char DEL=83;

const char END=79;

const char PAGDOWN=81;

const char CENTRE=76;

const char HOME=71;

const char PAGUP=73;

\*/

--------------------------------------------------------------------------

Lista dei comandi DOS

Post n°13 pubblicato il 02 Aprile 2010 da C\_piu\_piu

.exe - per avviare i file tipo EXE. Per avviare un file EXE scrivere: nome del file.exe

assign - reindirizza tutte le richieste a un drive, verso un altro drive

attrib - visualizza o modifica gli attributi dei file

backup - crea una copia di sicurezza di uno o più file su una serie di floppy-disk

break - abilita o disabilita l'interruzione o l'esecuzione fornita da CTRL C e CTRL BREAK

cd - lett. change directory, entra in una directory. Sintassi: "cd" per entrare in C, e "cdpercorso directory" per entrare in altra cartella.

chdir - cambia la directory corrente o ne visualizza il nome

chkdsk - analizza il contenuto del disco e visualizza una relazione sul suo stato

cls - cancella il contenuto della schermata (questo comando non è distruttivo)

color - permette di impostare il colore dello sfondo e del testo

command - avvia un nuovo interprete dei comandi MS-DOS

comp - confronta il con

copy - Copia uno o più file in un'altra posizione

ctty - cambia la console di input/output da cui vengono impostati i comandi

date - visualizza e permette di aggiornare la data

defrag - riorganizza i file in un disco rigido per ottimizzarne l'uso

del - cancella uno o più file

deltree - elimina una directory e tutti i file e le sottodirectory in essa contenute.

dir - visualizza il contenuto di una directory

diskcopy - copia il contenuto di un disco floppy in un altro disco floppy

diskcomp - confronta il contenuto di due dischetti dello stesso tipo

echo - abilita o disabilita l'eco da file batch

edit - avvia MS-DOS Editor per creare o modificare i file ASCII

edlin - avvia un editor di righe di testo. Per magg. informazioni (Prompt comandi e edlin/?)

emm386 - abilita o disabilita la memoria espansa di EMM386

erase - cancella uno o più file specificati

exe2bin - converte in formato binario un file eseguibile

exit - provoca l'uscita da un interprete di comandi secondario per ritornare a un interprete di comandi o ad un programma precedenti

expand - decomprime uno o più file compressi

fastopen - accelera l'apertura di un programma

fc - confronta il contenuto di due file

fdisk - configura un disco rigido creando le partizioni, per l'utilizzo di MS-DOS

find - ricerca una sequenza di caratteri all'interno di uno o più file

for - permette di eseguire in modo iterativo comandi di MS-DOS

format - formatta un disco per l'utilizzo di MS-DOS

goto - trasferisce l'esecuzione ad una determinata posizione di un file batch

graftabl - carica in memoria i caratteri ASCII non BIOS per i modi grafici

help - visualizza la guida di MS-DOS. ATTENZIONE: la guida è troppo lunga per essere visualizzata in una sola schermata, quindi per visualizzarla schermata per schermata digitare help|more (il simbolo "|" è il tasto alla sinistra tasto uno (1) altrimenti alt+124 = "|" e scrivere more

hexdump - lista in formato esadecimale il contenuto di un file, byte per byte

if - condiziona l'esecuzione di un comando in un file batch

ipconfig - visualizza ip attuale del computer

join - unisce un drive di disk ad una directory vuota su di un altro drive

keyb - configura la tastiera per la lingua specificata

label - crea, modifica o elimina l'etichetta di volume di un disco

md - crea una directory (si può utilizzare anche il comando mkdir, abbreviazione di "make directory")

mem - visualizza la quantità della memoria usata e quella libera del sistema

mode - stabilisce le modalità di visualizzazione e trasmissione per la stampante

more - permette la visualizzazione di informazioni, (files, pagine, o altri tipi di dati) che per la loro grandezza (insieme di righe) devono essere visualizzate con più schermate alla volta.

net share - permette ad un computer di condividere dei file in rete

net start - apre un servizio

net stop - chiude un servizio

path - visualizza o imposta un percorso di ricerca per file eseguibili

pause - provoca una pausa nell'esecuzione di un file batch per consentire un input da operatore

print - stampa un file di testo

prompt - cambia il prompt dei comandi di MS-DOS

rasdial - seguito da /disconnect disconnette l'ADSL

rd - cancella una directory

recover - recupera un file o un intero disco con settori danneggiati

reg add - aggiunge chiave di registro

reg delete - elimina chiave di registro

rem - visualizza commenti durante l'esecuzione di un file batch

ren - rinomina uno o più file

replace - sostituisce i file

restore - ripristina dai dischi di back-up un certo numero di file

rmdir - rimuove una sottodirectory vuota

select - copia il dischetto MS-DOS per creare una copia di lavoro per la nazionalità selezionata della tastiera

set - assegna un valore di sostituzione ad un parametro con chiave presente in un programma applicativo o in un file batch

share - installa la condivisione ed il bloccaggo dei file MS-NET

shift - aumenta il numero di parametri sostituibili in un file batch

systeminfo - visualizza molte informazioni utili riguardanti le connessioni e altro

shutdown - arresta o riavvia il sistema

sort - ordina i dati alfabeticamente, in un senso o nell'altro

/stext - manda le informazioni di alcuni tipi di file eseguibili a file di testo

subst - associa un percorso ad una lettera di unità

sys - copia i file di sistema di MS-DOS e l'interprete dei comandi sul disco specificato

time - visualizza e consente di modificare l'ora

tracert - consente di tracciare il percorso sulla rete compiuto verso un sito/IP

tree - visualizza tutte le directory ed i percorsi del drive specificato

type - visualizza il contenuto di un file di testo

undelete - ripristina i file cancellati precedentemente con il comando del

unformat - ripristina i dati di un disco cancellati con il comando format

ver - visualizza la versione di MS-DOS in uso

verify - abilita la verifica dei dati in scrittura

vol - visualizza l'etichetta e il numero di serie di volume del disco

xcopy - copia i file e la struttura delle directory (ad esclusione dei file nascosti e di sistema)

@ - se messo all'inizio di una linea nasconde i dettagli durante la loro esecuzione