Laboratorio di Calcolo per Fisici, Prima prova valutata,

Canale Pb-Z

Lo scopo della prova di esame è scrivere un programma che calcoli l'area di diverse figure piane utilizzando un metodo *hit and miss*.
Per svolgere l'esercitazione avrete 3 ore; sono concessi libri di testo e appunti, ma **l'uso di cellulari, laptop e tablet non è ammesso, pena l'annullamento del compito.**
Tutti i file vanno scritti e salvati esclusivamente nella *home directory* dell'utente *studente*; il listato *c* deve essere salvato su un file di nome *gruppoXX.c*, mentre il file python e il grafico da esso generato si dovranno chiamare *gruppoXX.py* e *gruppoXX.png*. Per sicurezza inserite nelle prime righe del file in *c* due righe di commento contenenti il vostro nome, cognome e numero di matricola.

► **Esercizio:**
Il metodo *hit and miss* è un metodo stocastico che permette, con un ipotetico esperimento di tiro al bersaglio, di stimare l'area F di una certa figura piana contenuta all'interno di un quadrato di area nota A. In pratica si genera, all'interno del quadrato, una successione casuale di N punti P_i di coordinate x_i, y_i $(i = 1, 2, 3, ...N)$, e poi si contano i punti che colpiscono il bersaglio, cioè che cadono all'interno della figura. Se il numero totale di punti generati è N e il numero di punti che colpiscono il bersaglio è M, si può dimostrare che, se N è abbastanza grande,

$$\frac{M}{N} \to \frac{F}{A}, \quad \text{ovvero che} \quad F \to A \times \frac{M}{N} \cdot$$

Scopo finale dell'esercitazione è calcolare l'area colorata in bianco nel pannello (b) della figura sottostante. L'esercitazione è divisa in due parti. Nella prima si calcola l'area del cerchio di rggio R_1 raffigurato nel pannello (a). Nella seconda si calcola l'area della figura più complicata raffigurata nel pannello (b), ottenuta ritagliando (sottraendo) dal cerchio raffigurato nel pannello (a) un cerchio piú piccolo di raggio $R_2 < R_1$. Alla fine si chiede di generare un grafico dell'esperimento (vedi sotto).

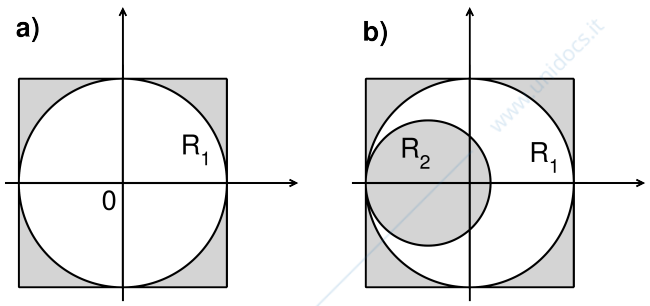

Figure 1: I valori numerici dei parametri in figura sono: raggio del cerchio grande: $R_1 = 2\,\text{cm}$ e $R_2 = 1.2\,\text{cm}$; lato del quadrato circoscritto al cerchio piu' grande ($L = 2 \cdot R_1 = 4\,\text{cm}$); centro del cerchio più grande ($x_1 = 0\,\text{cm}, y_1 = 0\,\text{cm}$); centro del cerchio più piccolo ($x_2 = -0.8\,\text{cm}, y_2 = 0\,\text{cm}$).

L'esercitazione va svolta come segue:

Prima parte:

1. Mediante una direttiva al precompilatore, si definiscono i valori dei parametri numerici definiti nella legenda della figura.
2. Il programma chiede quindi all'utente di inserire un numero N di tentativi – per avere un risultato sensato, si consiglia di scegliere un numero N maggiore di 10.000.
3. A questo punto, il programma esegue N volte le seguenti operazioni:
   - Attraverso una funzione `genconv` si genera in maniera casuale una coppia di coordinate (x_i, y_i), e le si salva in un vettore `pcoord` di tipo e dimensioni opportune.
   - Una funzione `isincircle` riceve in input le coordinate (x_i, y_i) e controlla se il punto corrispondente è contenuto all'interno del cerchio di raggio R_1, centrato nel punto x_1, y_1; la funzione deve essere scritta in maniera tale da funzionare per un cerchio generico di raggio R centrato nel punto (x_c, y_c).
   - In caso affermativo, conteggia l'evento come un *hit*.
4. Alla fine del ciclo, il programma stima l'area F del cerchio di raggio R_1 mediante la formula

$$F(R_1) = A \times \frac{M}{N}$$

Seconda parte:

1. Una volta verificato che l'algoritmo *hit and miss* è stato implementato correttamente, si modifichi il programma in modo da calcolare l'area in bianco nel pannello (b) della figura, riutilizzando opportunamente le funzioni scritte nella prima parte.
2. Alla fine del programma, si stampino sullo schermo le seguenti informazioni, con questo formato:
   `Numero di lanci:  10.000 | Area della figura a | Area della figura b`
3. Si salvino su un file *coppie.dat* le coordinate x_i, y_i dei punti contenuti nell'area di interesse, cioè i punti usati per stimare l'area del pannello (b) della figura.
4. Tramite uno script `python` si crei su un file *gruppoXX.png* un grafico che mostri le coordinate dei punti contenuti nell'area di interesse.

PROGRAMMA IN C ESEGUIBILE*

*__ATTENZIONE!__ Nota anche che, affinché la funzione fopen non dia il messaggio di errore, devi creare il file .txt** vuoto nella stessa cartella in cui hai creato il file.c.

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
#include<time.h>
#define R1 2      //cm
#define R2 1.2    //cm
#define L (2*R1)  //cm
#define A (L*L)   //cm^2  area quadrato
#define x1 0       //x centro cerchio 'grande'
#define y1 0       //y centro cerchio 'grande'
#define x2 (-0.8)   //x centro cerchio 'piccolo'
#define y2 0       //y centro cerchio 'piccolo'
#define TRY 10000     //generazione minima di numeri consigliata



//PROTOTIPI
void welcome(void);
int getN(void);
void azzera_bi(double c[][2], int d, int d1);
void genconv(double e[][2], double max, double min, int ind);
int isincircle(double f[][2], double xc, double yc, double r, int ind2);
double area(double aq, int M, int trys);
void end_message_A1(int n, double ac1, double ac2);



//FUNZIONE main
int main(){
  int N, i;
  int seed;
  int hit_a=0;
 int hit_b=0;
 double F1, F2, F;
 FILE* fp;


 seed = time(0); //inizializzazione del seed
 srand48(seed);



 //Chiamata Funzioni
welcome();
N = getN();
```

```
 double pcoord[N][2];

 azzera_bi(pcoord, N, 2);



 fp = fopen("coppie.dat", "w+");

 if((fopen("coppie.dat", "w+"))==NULL){        //**
printf("Errore apertura file.\n");
 exit(1);
 }


 for(i=0; i<N; i++){
  genconv(pcoord, R1, (-R1), i);
 if((isincircle(pcoord, x1, y1, R1, i))==1) { hit_a++; };
 if((isincircle(pcoord, x2, y2, R2, i))==1) { hit_b++; }

 if(isincircle(pcoord, x1, y1, R1, i)==1 && isincircle(pcoord, x2, y2, R2, i)==0) {
  fprintf(fp, "%7.6lf\t\t%7.6lf\n", pcoord[i][0], pcoord[i][1]);
}
}


 fclose(fp);


 F1 = area(A, hit_a, N);
 F2 = area(A, hit_b, N);
 F = F1 - F2;
end_message_A1(N, F1, F);


 return(0);
}




//FUNZIONE welcome
void welcome(){

  printf("\nBenvenuto.\nQuesto programma stima l'area data dalla differenza fra le aree di
due cerchi inscritti in un quadrato di lato %d.\n\n", L);
}




//FUNZIONE getN
int getN(){
  int a;

  printf("Quanti numeri vorresti generare per il calcolo dell'area?\nN = ");

  do{
   scanf("%d", &a);
```

```
 if(a<TRY) printf("Si consiglia di generare almeno %d numeri.\nN = ", TRY);
 } while(a<TRY);


return(a);
}




//FUNZIONE azzera_bi
void azzera_bi(double c[][2], int d, int d1){
  int i, j;

 for(i=0; i<d; i++){
  for(j=0; j<d1; j++){
  c[i][j] = 0;
  }
}


}




//FUNZIONE genconv
void genconv(double e[][2], double max, double min, int ind){
  double x=0;
  double y=0;


  x = ((double)lrand48()/RAND_MAX) * (max-min) + min;
   e[ind][0] = x;

 y = ((double)lrand48()/RAND_MAX) * (max-min) + min;
   e[ind][1] = y;


}




//FUNZIONE isincircle
int isincircle(double f[][2], double xc, double yc, double r, int ind2){
 int g=0;

 if(pow(f[ind2][0]-xc, 2) + pow(f[ind2][1]-yc, 2) <= r*r){
  g = 1;
 }

 return(g);
}
```

```
//FUNZIONE area
double area(double aq, int M, int trys){
  double ac=0;

ac = aq * (double)M/trys;

return(ac);
}




//FUNZIONE end_message_A1
void end_message_A1(int n, double ac1, double ac2){

 printf("\nNumero lanci:\t%d\t|\tArea figura a:\t%lf cm^2\t|\tArea figura b:\t%lf cm^2.\n\n",
n, ac1, ac2);

}
```

SCRIPT IN PYTHON

```
#This Python file uses the following encoding: utf-8

import matplotlib.pyplot as plt
import numpy as np

x,y=np.loadtxt('coppie.dat', unpack=True, usecols=(0, 1))
plt.title('Area figura b con metodo hit or miss')
plt.xlim(-2, 2)
plt.ylim(-2, 2)
plt.xlabel('coordinate x')
plt.ylabel('coordinate y')
plt.plot(x,y,'mo', label='area b', linewidth=3)
plt.legend()
plt.savefig('exam_8.png')
plt.show()
```

ALLEGATO DI COME DOVREBBE USCIRTI IL GRAFICO