INFORMATICA A – Prof. Fuggetta

Laboratorio 2

Ing. Paolo Perego, Ph.D.

Ing. Anna Maria Nestorov, MSc.

Esercizio 1
Array

Scrivere un programma che calcola l'approssimazione di π, grazie a una funzione *pi* avente come unico parametro la sua precisione (intero < 9 inserito dall'utente), calcolata come:

$$\sum_{k=0}^{\texttt{precision}} \frac{1}{16^k}\left(\frac{4}{8k+1} - \frac{2}{8k+4} - \frac{1}{8k+5} - \frac{1}{8k+6}\right)$$

L'approssimazione di π calcolata usando la precedente formula con la precisione inserita, deve avere 30 cifre decimali dopo la virgola.

Esempio:
Precisione: 6
Approssimazione: 3.14159265322808778364560566842 6

Esercizio 2

Dati in ingresso massimo 20 numeri (ma possono essere anche meno, terminare la lettura inserendo un qualsiasi numero negativo), scrivere un programma che, attraverso funzioni, permetta di:

1. Calcolare la somma di tutti i numeri inseriti, il minimo ed il massimo;
2. Calcolare la media (usando la funzione precedente), escludendo il numero massimo e il numero minimo
3. Restituire solo i numeri dispari (non usare i puntatori)
4. Determinare se il numero è primo o meno (restituisce 1 se primo, 0 se non primo)
5. Restituire solo i numeri primi (nuovo array).

Esercizio 3

Scrivete un programma che legga, utilizzando un'array, una stringa di al massimo 100 caratteri formata solo da lettere minuscole e dallo spazio e terminata da '$'.

Quindi:

1. Per ogni carattere della stringa, mettere il carattere alfabeticamente successivo;
   * il carattere successivo della 'z' sarà la 'a';
   * il carattere successivo dello spazio, sarà '_'.

   Per esempio, 'a' → 'b' , 'g' → 'h', 'z' → 'a', ' ' → '_'.
2. Stampare la stringa così codificata.

(Assumere che la stringa sia composta effettivamente di soli caratteri minuscoli e di spazi)

Esercizio 4

Scrivete un programma che generi a caso due array di N=10 elementi interi, il cui valore sia compreso tra 1 a 200.

Si richiede di:

- Creare una funzione per generare gli array;
- Creare una funzione che stampi gli array generati;
- Creare una procedura che costruisca un terzo array di dimensione 2N i cui elementi di posizione pari siano gli elementi del primo vettore e gli elementi di posizione dispari siano gli elementi del secondo vettore

Per generare numeri casuali usare la funzione rand()

```
#include <time.h>
#include <stdlib.h>
…
int main(){
srand(time(NULL));    //solo all'inizio del main
int r = rand();  //genera un int casuale da 0 a RANDMAX
int g = rand() % 20;  //genera un int casuale da 0 a 19
```

Esercizio 5

Si sviluppi una funzione filtro che, dati in input:

- la stringa di 38 caratteri *“Nel mezzo del cammin di nostra vita...”*,
- un array di valori booleani avente il valore 1 (filtra) solo se l’indice è 0 o multiplo di 3, altrimenti 0 (non filtrare)

restituisca una nuova stringa filtrata contenente:

- in corrispondenza del valore 1, il carattere della prima stringa;
- in corrispondenza del valore 0 degli spazi.

Esercizio 6

Scrivere un programma che, dato un vettore *vett_in* di 10 interi che:

1. Tramite una funzione, generi un secondo array *vett_out_molt* così definito:
   - l'elemento 0 è dato dal prodotto degli elementi 0 e 1 di *vett_in*,
   - l'elemento 1 è dato dal prodotto degli elementi 2 e 3 di *vett_in*, e così via.

   Esempio:
   vett_in [] = {1,2,3,4,5,6,7,8,9,10} -> *vett_out_molt* [] = {2, 12, 30, 56, 90}

2. Tramite una funzione, generi un terzo array *vett_out_somma* così definito:
   - l'elemento 0 è dato dalla somma degli elementi 0 e 9 di *vett_in*,
   - l'elemento 1 è dato dalla somma degli elementi 2 e 8 di *vett_in* , e così via.

   Esempio:
   vett_in [] = {1,2,3,4,5,6,7,8,9,10} -> *vett_out_somma*[] = {11, 11, 11, 11, 11}

3. Tramite una funzione, calcoli la percentuale di valori positivi e di valori negativi presenti in *vett_in*

Si supponga che il numero di elementi di *vett_in* sia pari.

Esercizio 7

Si scriva un programma che manipoli un array c di char tale che:

L'array deve avere valore iniziale

['A' 'B' 'c' 'D' 'e' 'f']

E, dopo aver applicato due funzioni "toUpperCase" e "toLowerCase", questo contega:

['a' 'b' 'C' 'd' 'E' 'F']

Esercizio 8

Si scriva una funzione in linguaggio C che prenda in ingresso il seguente array di 22 interi

```
int array_in[DIM] = {67, 79, 77, 80, 76, 73, 77, 69, 78,
                     84, 73, 33, 72, 97, 105, 70, 105,
                     110, 105, 116, 111, 33};
```

ed un puntatore ad un secondo array di char. La funzione riempie il secondo array con i valori del primo array convertiti nel rispettivo carattere (secondo il codice ASCII). Non utilizzare gli indici per gli array, ma solo la logica dei puntatori.