

Compiti Natale 3AIIN 2020

Caricare tutti i lavori su Moodle!

1) Ripassare/studiare tutte le slide viste in classe, gli appunti e il libro di testo. Sul registro elettronico trovi indicati tutti gli argomenti svolti nel trimestre.

2) Rivedi le prove scritte, gli esercizi svolti in classe, quelli assegnati per casa e in laboratorio.

3) STUDIA LA TEORIA!!!

Mi raccomando:

- **prima si analizza/progetta la soluzione (algoritmo) poi si realizza (codifica) il programma.**
- utilizza nomi significativi per costanti e variabili
- utilizza le costanti
- rispetta le convenzioni viste per gli identificatori (di costanti e variabili)
- indenta con la massima cura l'algoritmo/pgm (come visto in classe)
- separa con una riga vuota i diversi blocchi logici dell'alg./pgm.
- abituati ad inserire come commento la funzione svolta dal programma e ad inserire commenti a lato di istruzioni o parti di codice che ritieni necessari
- **abituati a testare il pgm dopo aver fatto un piano di test. Se qualche cosa non funziona fai la trace come ti ho insegnato.**

Esercizio n.1

Scrivi un algoritmo (utilizza il linguaggio di progetto) che legge una sequenza di numeri interi positivi arrestandosi quando la somma dei numeri immessi supera il primo valore che viene immesso.

L'algoritmo restituisce la quantità dei numeri letti.

Rispetta tutte le regole, convenzioni e indicazioni date in classe.

Esempi:

13 3 5 2 1 2 5 → 6

20 45 → 1

Scrivi **poi** il programma.

Esercizio n.2

Una pianta cresce ogni mese della metà di quanto è cresciuta il mese precedente (il primo mese cresce della metà dell'altezza iniziale).

Progettare un algoritmo, utilizza il linguaggio di progetto, che calcola l'altezza finale della pianta a partire dall'altezza iniziale e dal numero di mesi.

Esegui un controllo dell'input.

Rispetta tutte le regole, convenzioni e indicazioni date in classe.

Esempio:

altezza iniziale: 20

numero mesi: 3

1° mese cresce di $20/2 = 10$

2° mese cresce di $10/2 = 5$

3° mese cresce di $5/2 = 2.5$

Dopo 3 mesi la pianta è alta:

$20 + 10 + 5 + 2.5 = 37.5$

Scrivi **poi** il programma.

Esercizio n.3

Scrivi un algoritmo (utilizzando il linguaggio di progetto o pseudolinguaggio) che letto un numero x maggiore di 1 determini il più piccolo intero k tale che

$$2^k \geq x$$

Esegui un controllo dell'input.

Non puoi utilizzare funzioni per il calcolo della potenza.

Rispetta tutte le regole, convenzioni e indicazioni date in classe

Scrivi **poi** il programma.

Esercizio n.4

Determinare l'output del seguente frammento di codice C++:

```
int a = 12;
    while ( a != 1 )
        cout<<(a = (a%2) ? (a*3 + 1) : (a/2)))<<"\t";
```

Data la seguente espressione logica scritta in C++, dire per quali casi, ossia per quali valori, risulti vera o falsa, senza scrivere la tavola di verità.

!(a != c) || ((a > c) && (c > a))

Riscrivi (su carta!) il seguente frammento di codice C++ usando solo cicli while():

```
int i, j, s;
s = 0;
for (i = 10; i > 4; i--) {
    j = i;
    do{
        s = s + j;
        j = j + 2;
    }while (j < 10);
}
```

Indicare poi il valore delle variabili i, j s al termine dell'esecuzione del frammento di codice.

I due frammenti di codice provali poi su computer per verificare l'equivalenza e la correttezza del contenuto delle variabili i, j s al termine dell'esecuzione del frammento di codice.

Riscrivi (su carta!) il seguente frammento di codice C++ usando solo cicli while. Determina poi quanto vale la variabile s al termine dell'esecuzione del codice.

```
s=0;
for(i=0; i<5; i++){
    d=1;
    for(j=0; j<i; j++)
        d+=2;
    s+=d;
}
```

I due frammenti di codice provali poi su computer per verificare l'equivalenza e la correttezza del contenuto della variabile s al termine dell'esecuzione del frammento di codice.

Esercizio n.5

Scrivi un pgm C++ che calcola il minimo comune multiplo tra due numeri m ed n (interi positivi) procedendo nel seguente modo: confrontare i due valori e sommare se stesso al valore più piccolo, confrontare la somma ottenuta con l'altro valore, sommare un altro valore alla somma più piccola finché diventano uguali le due somme (osserva bene l'esempio!).

Esegui un controllo dell'input.

Es. $m = 3, n = 5$

$(3+3, 5)$ $(6, 5+5)$ $(6+3, 10)$ $(9+3, 10)$ $(12, 10+5)$ $(12+3, 15)$ $(15, 15)$

quindi $\text{mcm}(3,5) = 15$

Esercizio n.6

Scrivi un pgm C++ che legge una sequenza di 0 e 1 terminante con il numero -1 (non effettuare alcun controllo dell'input) e determina qual è la più lunga sequenza di 1 consecutivi letti, la posizione iniziale e quella finale.

Esempio:

Input:

0 0 1 1 0 1 1 0 0 0 0 **1 1 1** 0 -1

Output:

Lunghezza = 3

Inizio = 12

Fine = 14

Esercizio n.7

Un numero di Lynch-Bell è un numero naturale divisibile per tutte le sue cifre (diverse da zero). Ad esempio, 1236 è un numero Lynch-Bell perché è divisibile per 1, 2, 3, e 6. Anche 1000 è un numero di Lynch-Bell, perché è divisibile per l'unica cifra non zero (1).

Scrivere un programma C++ che stampa tutti i numeri di Lynch-Bell compresi tra inf e sup $[\text{inf}, \text{sup}]$, estremi inclusi, con $\text{inf} < \text{sup}$.

Esegui opportuni controlli di input.

Esercizio n.8

Scrivere un algoritmo (utilizza il linguaggio di progetto) che risolva il seguente problema: dati due numeri interi positivi N e K , chiedere all'utente di inserire altri N numeri interi positivi $< K$ e verificare quanti di questi numeri sono un suffisso per K , stampandone successivamente la quantità. Un numero num , composto da C cifre, è un suffisso di K se le ultime C cifre di K sono le stesse di num , nello stesso ordine.

Ad esempio: 684 è un suffisso di 123684, mentre 685 non lo è.
Esegui opportuni controlli di input.

Scrivi **poi** il programma.

Esercizio n.9

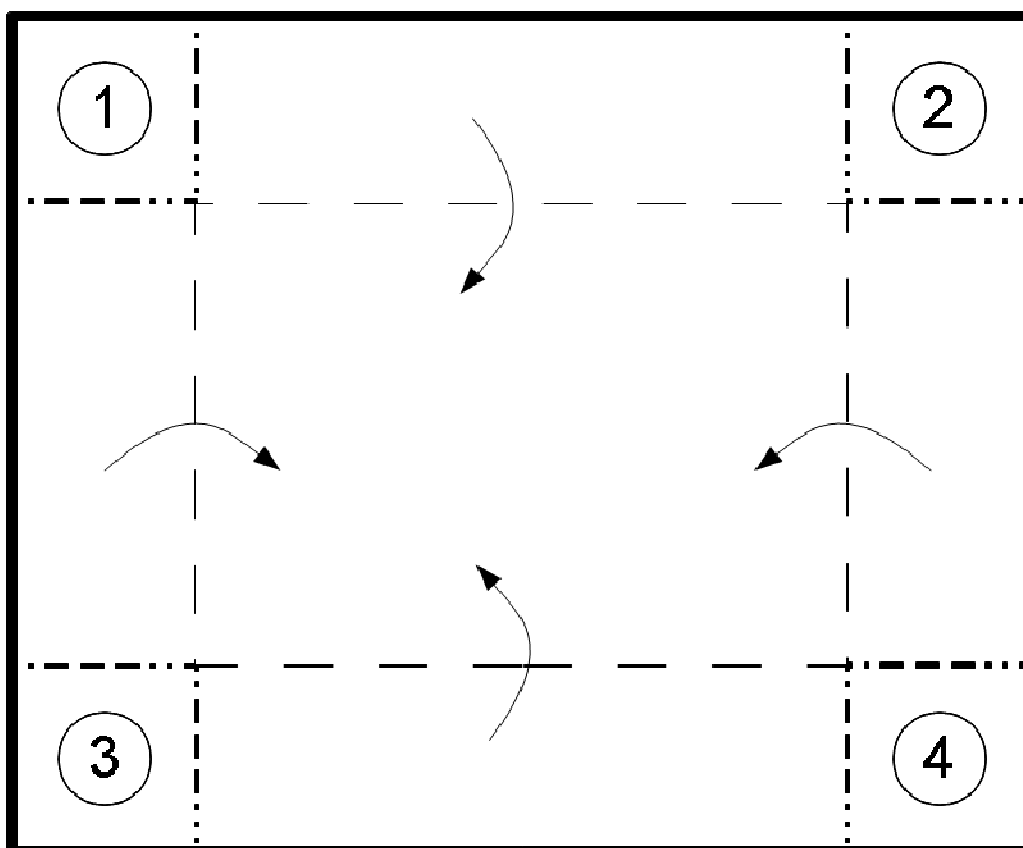
Scrivere un programma C++ che presi in ingresso due valori L e R, stampi R quadrati di lato L, senza la base, come nel disegno sottostante in cui i valori letti sono L = 4 e R = 3 (la distanza tra un quadrato e il successivo è uguale a L). Si supponga che sia L che R siano strettamente positivi e in particolare che $L \geq 2$.

Xxxx	xxxx	xxxx
x x	x x	x x
x x	x x	x x
x x	x x	x x

Esercizio n.10

Problema del taglio di una lastra di metallo per ottenere un contenitore di volume massimo date le dimensione della lastra.

Si supponga di avere un macchinario che data una lastra di metallo di forma rettangolare tagli un quadrato da ognuno degli spigoli, in modo che poi la si possa piegare e facendo combaciare i lati così ottenuti si possa ottenere un parallelepipedo, come si può vedere dalla figura. A seconda della dimensione dei lati dei quadrati ritagliati (indicati in figura con 1,2 3 e 4) il volume risulta diverso, si vuole trovare il lato del quadrato che permette di massimizzare il volume del parallelepipedo.



Caricare tutti i lavori su Moodle!