

Programmazione I

Prova scritta - 11 giugno 2012 - 1h20min

NOTA: Nei programmi si trascuri ogni problema legato al tipo ed al valore di ritorno della funzione **main**, inoltre si sottintenda la presenza delle direttive

```
#include <iostream> / #include <fstream> / using namespace std ;
```

e non si prenda come un buon esempio la formattazione utilizzata (spesso compressa per motivi di spazio). Si interpreti “terminazione forzata”, come l'abbreviazione di “terminazione forzata del programma da parte del sistema operativo”. Infine, laddove si trovi l'affermazione che un programma o frammento di codice produce un certo risultato, è da intendersi che, in accordo alle regole del linguaggio, tale programma o frammento di codice produce quel risultato per **qualsiasi esecuzione** su **qualsiasi macchina**.

PARTE 1 – RISPOSTA SINGOLA - Ogni domanda ha una sola risposta VERA.

- Una risposta esatta fa acquisire il punteggio positivo riportato a fianco della domanda
- Una risposta errata fa perdere il punteggio negativo riportato a fianco della domanda
- Una risposta lasciata in bianco viene valutata 0

1. (3, -0.5) Dato il seguente programma e supponendo che l'utente inserisca sullo *stdin* una sequenza, di lunghezza arbitraria, di caratteri non separati da spazi, dopodiché segnali la fine dell'input con, ad esempio, Ctrl+D da terminale Unix:

```
main() { char str[10], c; int i= 0 ;  
        while(cin>>c) str[i++] = c ;  
        str[i] = '\0'; cout<<str ;}
```

- a) se eseguito il programma stampa la sequenza di caratteri inserita dall'utente
- b) nessuna delle altre risposte è vera
- c) se eseguito il programma non stampa nulla
- d) il programma stampa la sequenza di caratteri inserita dall'utente, seguita dal carattere speciale di codice 0

2. (3, -0.5) Il seguente programma

```
main()  
{  
    for (int i = 1 ; i < 2 ; i++)  
        for (int j = i ; j < 3 ; j++) {  
            cout<<i<<" "<<j<<" " ;  
            if (j % 1 == 0)  
                j++ ;  
        }  
}
```

- a) stampa 1 1
- b) stampa 1 1 2 2
- c) stampa 1 1 2 2 3 3
- d) nessuna delle altre risposte è vera

3. (3, -0.5) Dato il seguente programma:

```
main() {
    int i ; cin>>i ; unsigned int u = 25 ;
    cout<<u+static_cast<unsigned int>(i) ;
}
```

- per qualsiasi numero intero, positivo o negativo, immesso dall'utente, il programma stampa un numero non negativo
- per qualsiasi numero intero i , positivo o negativo, immesso dall'utente, il programma stampa il valore dell'espressione $25 + i$
- se l'utente immette -25 da *stdin*, il programma viene terminato forzatamente
- nessuna delle altre risposte è vera

4. (3, -0.5) Data la seguente funzione a cui viene passato un array a di dimensione N :

```
bool fun(unsigned int a[], int i, unsigned int N) {
    if ( (i < 0 && i >= N) || a[i] % 2 == 1 )
        return false ;
    return true ;
}
```

- Solo se l'indice i è compreso tra 0 ed $N-1$ (estremi inclusi) la funzione controlla il valore dell'elemento i -esimo e ritorna **false** se tale elemento è pari
- Solo se l'indice i è compreso tra 0 ed $N-1$ (estremi inclusi) la funzione controlla il valore dell'elemento i -esimo e ritorna **false** se tale elemento è dispari
- Nessuna delle altre risposte è vera
- La funzione può scrivere al di fuori dell'array a

PARTE 2 – (POSSIBILI) RISPOSTE MULTIPLE -

Ogni domanda può avere da una a quattro risposte CORRETTE.

- Ogni risposta esatta viene calcolata: +1
- Ogni risposta errata viene calcolata: -0.5
- Una risposta lasciata in bianco viene calcolata: 0

5. Data una sequenza di N valori interi positivi

- La struttura dati più efficiente, in termini di tempo di lettura di tutti i valori, in cui si potrebbero memorizzare tali valori è un *array* di N elementi
- La struttura dati più efficiente, in termini di occupazione di memoria, in cui si potrebbero memorizzare tali valori è una lista semplice di N elementi
- Aggiungere un elemento nel mezzo della sequenza ha costo $O(N)$ se la sequenza è implementata mediante un *array*
- Se la sequenza è memorizzata in una lista semplice e senza memorizzare in alcuna altra variabile aggiuntiva il numero di elementi della sequenza, allora sono necessarie più di $O(1)$ operazioni per calcolare tale numero di elementi

6. Dato il seguente programma:

```
const int N = 1000 ;
struct ss { int b; char a[N] ; } ;
void fun(ss d) { d.a[N/2] = 1 ; d.b = 3 ; cout<<d.b ; }
main()
{
    ss c ; c.b = 13 ; c.a[N/2] = 2 ;
    fun(c) ; cout<<" "<<c.b<<" "<<static_cast<int>(c.a[N/2]) ;
}
```

- a) Quando **fun** è invocata il contenuto del parametro attuale **c** è copiato nel parametro formale **d**
- b) In quanto al campo **a** del parametro attuale **c** passato alla funzione **fun**, l'*array* in esso contenuto è copiato nel campo **a** del parametro formale **d**.
- c) Un passaggio per riferimento sarebbe stato più conveniente in termini di tempo di esecuzione
- d) Il programma stampa: **3 13 2**

7. La memoria occupata da un oggetto **X** allocato dinamicamente all'interno di una funzione
- a) è allocata all'inizio dell'esecuzione della funzione;
 - b) è liberata se si assegna l'indirizzo di un nuovo oggetto dinamico al puntatore che contiene l'indirizzo di **X** in memoria;
 - c) è liberata se viene deallocato il puntatore che contiene l'indirizzo di **X** in memoria;
 - d) se non esplicitamente deallocata, è liberata solo al termine del programma.

8. Dato il seguente programma:

```
main()
{
    ofstream f("nome.txt") ; // apro il file in scrittura
    f<<1<<2<<endl ;
    f<<3 ;
    ifstream f2("nome.txt") ; // riapro lo stesso file in lettura
    int i ;
    while(f2>>i) ;
        cout<<i<<" " ;
}
```

- a) al termine del programma, il file **nome.txt** è un file binario contenente i numeri 1, 2 e 3
- b) al termine del programma, il file **nome.txt** è un file di testo contenente numeri diversi da 1, 2 e 3
- c) il programma stampa almeno **1, 2,**
- d) quando viene riaperto in lettura alla creazione dell'*ifstream* **f2**, il file **nome.txt** può non contenere tutti i caratteri che sono stati immessi nello *ofstream* **f**

PARTE 3 – DOMANDE APERTE –

- Una risposta esatta fa acquisire il punteggio positivo riportato a fianco della domanda
- Una risposta errata può eventualmente causare una penalità che dipende dalla gravità dell'errore
- Una risposta lasciata in bianco viene calcolata: **0**

9. (4 pt) Cosa scrive su *stdout* il seguente programma?

```
void fun(int v[], int N) {
    for (int i = 0 ; i < N - 1 && v[i] < v[i+1] ; i++)
        v[i+1] = v[i] ;
}

main() {
    const int N = 4 ; int v[N] = {2, 4, 3, 1} ;
    fun(v, N) ;
    for (int i = 0 ; i < N ; i++)
        cout<<v[i] ;
}
```

10. (6 pt) Scrivere una funzione che prenda in ingresso una stringa contenente solo caratteri alfabetici o spazi, e ritorni: 1) il numero di parole (ossia sequenze di caratteri non separati da spazi) contenute nella stringa e 2) una nuova stringa contenente solo l'ultima parola contenuta nella stringa in ingresso.

Programmazione I

Prova scritta - 11 giugno 2012

Nome: _____ Cognome: _____

Matricola: _____

Indicare le risposte corrette apponendo una croce nella casella corrispondente. Per superare la prova bisogna aver raggiunto almeno 9 punti nelle domande a risposta singola/multipla, ed almeno 15 complessivamente. Questa è l'unica pagina che dovete consegnare. Per comodità avete anche un copia di questa pagina per calcolare il voto da sole/soli durante la correzione.

	Risposte				Punti/ Penalità	
	A	B	C	D		
1					3	-0,5
2					3	-0,5
3					3	-0,5
4					3	-0,5
5						
6						
7						
8						

Risposta alla domanda 9 (4 pt):

Risposta alla domanda 10 (6 pt):

Programmazione I

Prova scritta - 11 giugno 2012

Nome: _____ Cognome: _____

Matricola: _____

Indicare le risposte corrette apponendo una croce nella casella corrispondente. Per superare la prova bisogna aver raggiunto almeno 9 punti nelle domande a risposta singola/multipla, ed almeno 15 complessivamente. Questa è l'unica pagina che dovete consegnare. Per comodità avete anche un copia di questa pagina per calcolare il voto da sole/soli durante la correzione.

	Risposte				Punti/ Penalità	
	A	B	C	D		
1					3	-0,5
2					3	-0,5
3					3	-0,5
4					3	-0,5
5						
6						
7						
8						

Risposta alla domanda 9 (4 pt):

Risposta alla domanda 10 (6 pt):