

Programmazione I

Prova scritta - 1 febbraio 2010 - 1h30min

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 calcolata: 0

1. (3, -0.5) Dato il seguente programma:

```
int c ;
int precedente() { return c-- ; }
```

```
main() {
    cin>>c ;
    int d = precedente() ;
    int e = precedente() ;
    if (e < d) cout<<"Minore" ;
}
```

- a) per qualsiasi numero intero immesso dall'utente, il programma stampa sempre **Minore**
- b) il programma contiene un errore di gestione della memoria
- c) nessuna delle altre risposte è vera
- d) per qualsiasi numero intero negativo immesso dall'utente, il programma non stampa nulla

2. (3, -0.5) Dato il seguente programma:

```
main()
{
    int a ; cin>>a ; bool b = a > 1 ;
    if ( (a > 0) == (! b) )
        cout<<a ;
}
```

- a) Se l'utente immette **1** su *stdin* il programma stampa **1**
- b) Il programma contiene un errore di gestione della memoria
- c) Nessuna delle altre risposte è vera
- d) Qualsiasi valore l'utente immetta su *stdin*, il programma non stampa nulla

3. (1, -.5) Il seguente frammento di codice

```
double b;
b = 5/2 ;
```

- a) causa un errore a tempo di compilazione;
- b) assegna il valore 2 alla variabile **b**;
- c) assegna il valore 2.5 alla variabile **b**;
- d) causa un errore a tempo di esecuzione.

4. (3, -0.5) Dato il seguente programma:

```
void fun(int &) ;
int global ;
main() { int &ref = global; ref = 3 ; fun(ref) ; }

void fun(int &local)
{
    local++ ;
    cout<<local<<" "<<global ;
}
```

- Se eseguito, il programma stampa 4 3
- Se eseguito, il programma stampa 4 4
- Nessuna delle altre risposte è vera
- Il programma causa un errore a tempo di compilazione

5. (3, -0.5) Il seguente programma

```
main()
{
    for (int i = 1 ; i < 3 ; i++)
        for (int j = i ; j < 4 ; j++) {
            cout<<i<<" "<<j<<" " ;
            if (j % 2 == 0)
                j++ ;
        }
}
```

- stampa 1 1 1 2 2 2
- stampa 1 1 1 2 2 3
- stampa 1 1 1 2 2 3 2 2 3 3
- nessuna delle altre risposte è vera

PARTE 2 – (POSSIBILI) RISPOSTE MULTIPLE - Ogni domanda può avere una o più risposte CORRETTE.

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

6. Dato un programma che, oltre a svolgere altri eventuali compiti, effettua operazioni di inserimento/estrazione di singoli elementi, nonché stampa di tutti gli elementi, su di una sequenza, potenzialmente molto lunga, di elementi tutti dello stesso tipo:

- Se le operazioni più frequenti sulla sequenza sono inserimenti ed estrazioni in/dalla testa ed in/dal fondo, allora la struttura dati più conveniente da utilizzare, in termini di costo computazionale, per implementare la sequenza è un lista
- Se, all'inizio della propria esecuzione, il programma è informato prima del numero effettivo di elementi della sequenza e poi del contenuto di ciascuno di tali elementi, e tale numero di elementi non varierà più durante l'esecuzione programma, allora, in termini di costo computazionale ed occupazione di memoria, la struttura dati più conveniente da utilizzare per implementare la sequenza è un array dinamico
- Se il numero di elementi che saranno inseriti nella sequenza è noto a tempo di scrittura del programma e non avverranno mai estrazioni, allora la struttura dati più conveniente da utilizzare, in termini di semplicità di gestione, per implementare la sequenza è un array dinamico
- Dato il numero di elementi N della sequenza in un dato istante e supponendo che la sequenza sia implementata mediante una lista, allora in tale istante la sequenza occupa una quantità di memoria maggiore di quella che occuperebbe se fosse rappresentata attraverso un array di N elementi

7. Dato il seguente programma:

```
const int N = 2000 ;
struct ss { int b; char a[2000] ; } ;
void fun(ss & d) { cout<<d.a[N/2]<<" "; d.b = 3 ; }
main()
{
    ss c ; c.b = 10 ; c.a[N/2] = '2' ; cout<<c.b<<" " ;
    fun(c) ; cout<<c.b ;
}
```

- Il passaggio di **c** per riferimento nel **main** permette a **fun** di modificare **c**
- Il passaggio di **c** per riferimento ha come costo la copia nel parametro formale **d** del contenuto di **c** stesso
- Il programma stampa **10 2 3**
- L'esecuzione di **fun(c)** ; ha effetti collaterali sulla porzione della funzione **main** che segue l'invocazione della funzione **fun** stessa

8. Dato il seguente programma:

```
int fun(int b[], int n)
{
    int s = 0 ;
    for (int i = 0 ; i < n ; i++) { s += b[i] ; }
    delete [] b ;
    return s ;
}

main()
{
    int *a = new int[2] ; a[0] = 3 ; a[1] = 4 ;

    cout<<fun(a, 2)<<" " ;
    for (int i = 0 ; i < 2 ; i++)
        cout<<a[i]<<" " ;
}
```

- il programma stampa **7 3 4**
- nella funzione **fun**, all'interno del ciclo **for** non si modifica l'array dinamico allocato con l'operatore **new** all'inizio della funzione **main**
- quando l'esecuzione arriva all'inizio del ciclo **for** nella funzione **main**, il puntatore **a** non punta ad un oggetto correttamente allocato in memoria
- il programma contiene un errore di gestione della memoria

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 stampa su *stdout* il seguente programma?

```
int fun(char s[])
{
    int n = 0;
    for (int i = 0 ; s[i] != '\0' ; i++)
        if (isupper(s[i])) { // vero se s[i] è una lettera maiuscola
            s[i] = tolower(s[i]);
            ++n;
        } else
            cout<<s[i] ;
    cout<<endl ;
    return n;
}

int main()
{
    int f = 0;
    char stringa[] = "EsameInfo";
    cout<<stringa<<" "<<f<<endl ;
    f = fun(stringa);
    cout<<stringa<<" "<<f<<endl ;
}
```

10. **(6 pt)** Scrivere una funzione che prende in ingresso una stringa e ritorna una nuova stringa in cui ogni sequenza di più di due caratteri uguali contenuta nella stringa in ingresso è stata sostituita da due sole occorrenze di tale carattere. Ad esempio, presa in ingresso la stringa "aabhhhhaggf", la funzione deve ritornare la stringa "aabhhaggf". L'array di caratteri in cui è memorizzata la stringa ritornata deve avere le dimensioni minime necessarie per contenere tale stringa.

Programmazione I

Prova scritta - 1 febbraio 2010

Nome: _____ Cognome: _____

Matricola: _____ Corso di Laurea: _____

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					1	-0,5
4					3	-0,5
5					3	-0,5
6						
7						
8						

Risposta alla domanda 9 (4 pt):

Risposta alla domanda 10 (6 pt):

Programmazione I

Prova scritta - 1 febbraio 2010

Nome: _____ Cognome: _____

Matricola: _____ Corso di Laurea: _____

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					1	-0,5
4					3	-0,5
5					3	-0,5
6						
7						
8						

Risposta alla domanda 9 (4 pt):

Risposta alla domanda 10 (6 pt):