

Programmazione I

Prova scritta - 14 dicembre 2009 - 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. (2, -.5) Date le istruzioni `cout<<'1'<<'2'<<endl ;` e `cout<<12<<endl ;` e supponendo che l'operatore di uscita sia configurato per stampare i numeri interi in notazione decimale
- a) Non è vero che entrambe le istruzioni immettono la stessa sequenza di codici carattere sullo *stdout*
 - b) Tali istruzioni immettono sullo *stdout* la stessa sequenza di codici carattere dell'istruzione `cout<<1<<2<<endl ;`
 - c) Nessuna delle altre risposte è vera
 - d) La seconda istruzione immette su *stdout* un numero maggiore di byte rispetto alla prima
2. (3, -.5) Il seguente programma:
- ```
int a = 3 ;

int fun1(int b)
{ a += 1 ; b -= 2 ; return b ; }

void fun2(int &a) { a = fun1(a) ; }

main()
{
 int b = 5 ;
 fun1(b) ;
 fun2(b) ;
 cout<<a<<" "<<b<<endl ;
}
```
- a) contiene un errore di gestione della memoria
  - b) stampa 5 1
  - c) stampa 5 3
  - d) nessuna delle altre risposte è vera

3. (3, -.5) Assumendo 1) che nel seguente programma l'operazione di apertura del file in lettura abbia successo, 2) che nel file sia correttamente memorizzato l'array scritto dalla funzione **fun1** e 3) che il tipo **int** occupi 4 byte

```
void fun1(const int a[], int n){ ofstream f("nome");
 f.write(reinterpret_cast<const char *>(a),
 sizeof(int) * n) ; }
void fun2(int a[], int n){ ifstream f("nome");
 for (int i = 0 ; i < n ; i++)
 if (!(f>>a[i])) break ; }
main() {
 const int N = 3 ;
 int b[N] = {1, 4, 2}, c[] = {1, 1, 1};
 fun1(b, 3) ;
 fun2(c, 3) ;
 for (int i = 0 ; i < N ; i++) cout<<c[i]<<" " ;
}
```

- a) il programma si interrompe in un qualche punto prima di iniziare l'esecuzione del ciclo **for** all'interno del **main**  
 b) il programma contiene un errore di gestione della memoria  
 c) è possibile che il programma non stampi **1 4 2**  
 d) nessuna delle altre risposte è corretta
4. (2, -.5) La seguente istruzione:  
`cout<<static_cast<int>('c' - 'a' + '3' - '1') ;`  
 a) può causare problemi di *overflow*  
 b) immette il codice di un carattere non stampabile sullo *stream* di uscita standard (**cout**)  
 c) stampa **4**  
 d) nessuna delle altre risposte è corretta

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)
 i++ ;
 }
}
```

- a) stampa **1 1 1 2 2 3**  
 b) stampa **1 1 1 2 1 3**  
 c) stampa **1 1 1 2 2 3 2 2 3 3**  
 d) 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. Dati due algoritmi equivalenti:
- a) Non può mai accadere che per un dato ingresso uno dei due non termini, mentre l'altro termina in un numero finito di passi
  - b) Solo uno dei due potrebbe fornire l'uscita corretta a parità di dati in ingresso
  - c) I due algoritmi potrebbero risolvere lo stesso problema con un numero di passi molto diverso
  - d) I due algoritmi utilizzano le stesse strutture dati
7. Dato il seguente programma e tenendo conto di possibili problemi di *overflow*:
- ```
main(){unsigned int a, b, c ;
cin>>a>>b ; c = a - b ;
if (c < 10)
    cout<<"messaggio"<<endl ;}
```
- a) **c non può mai assumere un valore negativo**
 - b) Il programma potrebbe non stampare **messaggio** anche se **a < b + 10** quando si esegue l'istruzione **c = a - b ;**
 - c) Sia **a** che **b** che **c** non possono mai assumere un valore maggiore del valore massimo consentito per il tipo **unsigned int**
 - d) Il programma potrebbe non stampare **messaggio** anche se **c < 10** quando si esegue l'istruzione **if (c < 10)**
8. Dato un programma scritto in linguaggio C/C++
- a) L'esecuzione di una istruzione di allocazione dinamica della memoria non provoca nessuna variazione delle dimensioni del record di attivazione della funzione contenente tale istruzione
 - b) Il tempo necessario per inizializzare il contenuto di un record di attivazione aumenta all'aumentare del numero e delle dimensioni dei parametri formali
 - c) Per deallocare un record di attivazione è necessario reinizializzare il contenuto delle celle di memoria precedentemente occupate dal record stesso
 - d) Il tempo necessario per inizializzare il contenuto di un record di attivazione aumenta all'aumentare del numero e delle dimensioni delle variabili locali non inizializzate
9. Quali delle seguenti affermazioni sono vere?
- a) Un *i/ofstream* può essere associato solo ad un file di tipo testo
 - b) Uno *stream* è una sequenza di numeri da interpretarsi come caratteri nel caso di letture/scritture formattate
 - c) Dato un *file* di testo associato ad un *ofstream* **f** su cui si effettuano solo accessi sequenziali, ogni operazione di scrittura di una riga (terminata da *newline*) su **f** che ha successo si traduce nella scrittura della riga in fondo al file
 - d) Dato un *file* di testo associato ad un *ifstream* **f** e su cui si effettuano solo accessi sequenziali, ogni operazione di lettura di una sequenza di caratteri da **f** che ha successo comporta l'eliminazione degli stessi caratteri da **f** e quindi dal file

10. Dato il seguente programma:

```
int *fun(int n)
{
    int *b = new int[n] ;
    for (int i = 0 ; i < n ; i++)
        b[i] = i ;
    return b ;
}
```

```
main()
{
    int * const a = fun(2) ;
    for (int i = 0 ; i < 2 ; i++)
        cout<<a[i]<<" " ;
}
```

- a) il programma stampa 0 1
- b) la funzione **fun** ritorna l'indirizzo del primo elemento di un array che non muore alla fine dell'esecuzione della funzione stessa
- c) nessuna delle altre risposte è vera
- d) 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

11. (3 pt.) Cosa scrive su *stdout* il seguente programma?

```
struct st {int a, b ;} ;

int fun(st t, st &q)
{
    t = q ;
    q.a = t.b ;
    q.b = t.a ;
    return t.b ;
}

main()
{
    st p = {1, 2}, q = {3, 4} ;
    p.b = fun(p, q) ;
    cout<<p.a<<" "<<p.b<<endl ;
    cout<<q.a<<" "<<q.b<<endl ;
}
```

12. (6 pt) Scrivere una funzione che prenda in ingresso una stringa **s1**, e crei e ritorni una nuova stringa **s2** (precedentemente non esistente in memoria) costruita inserendovi come primo carattere **s1[0]** (ossia **s2[0] = s1[0]**), quindi aggiungendo il successivo carattere, tra quelli rimanenti di **s1**, che è uguale o in ordine alfabetico crescente rispetto ad **s2[0]**, poi aggiungendo il successivo carattere, tra quelli rimanenti di **s1**, che è uguale o in ordine alfabetico crescente rispetto ad **s2[1]**, e così via. Esempio: data la stringa “ghehfzgl”, la funzione deve ritornare la stringa “ghhz”.

Programmazione I

Prova scritta - 14 dicembre 2009

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					2	-0.5
2					3	-0.5
3					3	-0.5
4					2	-0,5
5					3	-0,5
6						
7						
8						
9						
10						

Risposta alla domanda 11 (3 pt):

Risposta alla domanda 12 (6 pt):

Programmazione I

Prova scritta - 14 dicembre 2009

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					2	-0.5
2					3	-0.5
3					3	-0.5
4					2	-0,5
5					3	-0,5
6						
7						
8						
9						
10						

Risposta alla domanda 11 (3 pt):

Risposta alla domanda 12 (6 pt):