

Programmazione I

Prova scritta - 27 febbraio 2013 - 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, -.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, -.5) Dato il seguente programma:
- ```
int fun(const int a[]) { return a[0] + 1 ; }
main() { int b[2] = {2, 3} ; cout<<fun(b) ; }
```
- a) Quando la funzione **fun** è invocata, il valore di ciascuno degli elementi dell'*array b* (parametro attuale) è copiato nel corrispondente elemento dell'*array a* (parametro formale)
 - b) Quando la funzione **fun** è invocata, l'indirizzo del (primo elemento) dell'*array b* (parametro attuale) è copiato nel parametro formale **a**
 - c) Se la funzione **fun** contenesse istruzioni che modificano il valore di un elemento dell'*array a* (parametro formale) sarebbe segnalato un errore a tempo di esecuzione, in particolare l'errore sarebbe segnalato nel momento in cui si tenta di eseguire tale istruzione
 - d) Nessuna delle altre risposte è vera

3. (3, -0.5) Dato il seguente programma e supponendo che l'operatore >> non modifichi il valore della variabile se l'oggetto **cin** è, oppure entra, in stato di errore:

```
main()
```

```
{ int i = 10 ; do {cin>>i ; cout<<i<<" , " ; } while(cin) ; }
```

se l'utente inserisce da *stdin* i caratteri **21** e **13**, separati da uno spazio, quindi preme Invio, ed infine preme CTRL+D (su sistema UNIX)

- il programma stampa su *stdout* **21, 13,**
- il programma stampa su *stdout* **21, 13, 10,**
- il programma stampa su *stdout* **21, 13, 13,**
- nessuna delle altre risposte è corretta

4. (2, -0.5) Data la seguente funzione a cui viene passato un array **a** di dimensione **N**, e facendo attenzione ad eventuali errori di gestione della memoria:

```
bool fun(unsigned int a[], int i, unsigned int N) {  
    if (a[i] % 2 == 1 || (i < 0 && i >= N))  
        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 **true** 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 **true** 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 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
- Accedere in lettura all'**i**-esimo elemento della sequenza ha costo $O(1)$ se la sequenza è implementata mediante una lista semplice e si conosce l'indirizzo dell'elemento in memoria
- Se la sequenza è implementata mediante una lista semplice, allora eliminare l'elemento di indice $N/2$ della sequenza ha costo $O(N)$ anche se si conosce l'indirizzo dell'elemento in memoria

6. Dato il seguente programma e facendo attenzione alle costanti letterali utilizzate:

```
main() {
    char c ; cin>>c ;
    switch(c) {
    case 0:
        cout<<"Primo" ;
    case '3':
        cout<<"Secondo" ;
        break ;
    default:
        cout<<"Errore" ;
    }
}
```

(nelle prime tre risposte si assume che il programma si compili correttamente)

- Se l'utente immette il carattere **0** su *stdin*, il programma stampa **Errore**
 - Se l'utente immette il carattere **3** su *stdin*, il programma stampa **Secondo**
 - Se l'utente immette le cifre del codice numerico usato per rappresentare il carattere **3** su *stdin*, il programma stampa **Secondo**
 - Il programma causa un errore a tempo di compilazione perché non si può utilizzare il tipo *char* in uno *switch*
7. Indicare quale/i delle seguenti espressioni booleane è/sono vera/e, nell'ipotesi che:
int a=0, b=2, c=1;
- $c || (a+b)$
 - $a || (b \ \&\& \ a > b+c \ \&\& \ c == 1)$
 - $a < b$
 - $!c \ \&\& \ (a < b)$
8. Dato un puntatore **int *p**, che punta ad un array di **int** allocato dinamicamente in memoria, l'istruzione: **p=0**;
- fa sì che **p** non punti più all'array;
 - se **p** era l'unico riferimento alla zona di memoria, impedisce di eliminare successivamente l'oggetto dalla memoria;
 - fa sì che la memoria allocata per l'array sia liberata in quanto non più utilizzata;
 - sposta l'array puntato da **p** all'indirizzo zero in memoria.

9. Dato il seguente programma

```
1: float a = 3.1 ;
2:
3: void fun(int a, int b)
4: {
5:     a += b ;
6: }
7: main()
8: { fun(a, 3) ; cout<<a<<endl ; }
```

- la variabile **a** definita alla riga 1 ha scope relativo a tutto il programma;
- la variabile **a** definita alla riga 1 ha tempo di vita pari all'intero programma;
- prima dell'assegnamento, la variabile **a** riferita alla riga 5 ha un valore diverso da 3.1;
- la variabile **a** definita alla riga 1 non è visibile alla riga 8

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**

10. **(5 pt.)** Descrivere il meccanismo del corto circuito logico in non più di sei righe.

11. **(6 pt)** Scrivere una funzione che prende in ingresso un oggetto di tipo struttura contenente un vettore dinamico di interi la cui lunghezza massima è nota a tempo di scrittura del programma, e, senza utilizzare memoria dinamica e senza modificare l'oggetto preso in ingresso, ritorna un oggetto struttura dello stesso tipo, in cui però il vettore di interi è memorizzato nel modo seguente: il primo elemento è uguale al primo elemento del vettore originale, mentre ciascuno degli elementi successivi è uguale alla differenza tra l'elemento con lo stesso indice nel vettore originale ed il minore tra gli elementi precedenti, sempre nel vettore originale. Ad esempio, preso in ingresso il vettore [5, 7, 2, 8], la funzione ritorna il vettore [5, 2, -3, 6].

Programmazione I

Prova scritta - 27 febbraio 2013

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					2	-0.5
5						
6						
7						
8						
9						

Risposta alla domanda 10 (5 pt):

Risposta alla domanda 11 (6 pt):

Programmazione I

Prova scritta - 27 febbraio 2013

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. Usate questa copia 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					2	-0.5
5						
6						
7						
8						
9						

Risposta alla domanda 10 (5 pt):

Risposta alla domanda 11 (6 pt):