Programmazione I
Prova scritta - 22 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)** Il seguente codice:

```
struct pers {char nome[25]; int anno;} ;
pers fun(pers p)
{ strcpy(p.nome, "andrea") ; p.anno = 1986; return p ; }
main()
{
 pers z = {"michele", 1985};
 pers y = fun(z) ;
 cout<<z.nome<<" "<<y.nome ;
 strcpy(z.nome, "mario") ; cout<<" "<<y.nome ;
}
```

   a) Stampa **andrea andrea mario**
   b) Stampa **michele andrea andrea**
   c) Stampa **andrea andrea andrea**
   d) Nessuna delle precedenti risposte è corretta

2. **(3, -0.5)** Dato il seguente programma e supponendo il codice del carattere *a* sia 97:

```
main() {
   char a = 'a' ; char b = 3 ;
   cout<<static_cast<int>(a+b) ;
}
```

   a) Il programma stampa **100**
   b) Il programma stampa **d**
   c) Il programma genera un errore a tempo di compilazione perché non si può effettuare la somma di due valori di tipo *char*
   d) Nessuna delle altre risposte è vera

3. **(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 (a[i] % 2 == 1 || (i < 0 && i >= N))
        return false ;
   return true ;
}
```

   a) 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
   b) 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
   c) Nessuna delle altre risposte è vera
   d) La funzione può scrivere al di fuori dell'array **a**

4. **(3, -.5)** Data la seguente stringa rappresentata mediante un array di caratteri,

```
char s[10] = "Soprano" ;
```

   le istruzioni:

```
s[2] = 't' ; s[3] = s[2] ; s[4] = 'o' ; s[5] = ' ' ;
```

   a) contengono uno o più errori di accesso alla memoria
   b) trasformano la stringa da "Soprano" a "Sotto"
   c) non modificano la lunghezza della stringa
   d) nessuna delle altre risposte è vera

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

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

5. Dato il seguente programma:

```
1: const int M = 10 ;
2: void fun(char a[], int n)
3:  {
4:     for (int i = n - 1 ; i < M ; i++)
5.        a[i] = ' ' ;
6:  }
7:  main()
8:  {
9:      char b[M] = "paolo" ;
10:     fun(b, 3) ;
11:     cout<<b ;
12: }
```

   a) il programma stampa **pa**
   b) la stringa memorizzata in **b** ha lunghezza 5 prima dell'invocazione della funzione **fun** alla riga 10
   c) l'istruzione alla riga 11 può stampare un numero di caratteri maggiore di 9
   a) alla riga 11 la variabile **b** contiene ancora una stringa correttamente rappresentata, anche se diversa da "paolo"

6. Dato un programma scritto in linguaggio C/C++
   a) Il tempo necessario per inizializzare il contenuto di un record di attivazione aumenta all'aumentare del numero e delle dimensioni dei parametri formali
   b) Il record di attivazione di ogni funzione contiene (oltre ad altre informazioni) le variabili globali a cui accede la funzione
   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 inizializzate

7. Data l'istruzione **f>>i ;** ove **i** è una variabile di tipo **int** precedentemente definita ed **f** è un oggetto di tipo **ifstream**, configurato per leggere numeri in base 10 ed associato ad un file correttamente aperto:
   a) Se, all'esecuzione di tale istruzione, non si è ancora raggiunta la fine del file il programma legge i byte del file a partire dalla posizione corrente, alla ricerca della rappresentazione decimale di un numero intero
   b) All'esecuzione di tale istruzione il programma si blocca sempre in attesa che l'utente invii dei caratteri sull'*ifstream*
   c) Se, all'esecuzione di tale istruzione, la porzione di file a partire dalla posizione corrente contiene una sequenza di caratteri che rappresenta un numero intero, seguita da uno spazio ed infine da una sequenza di caratteri che non rappresenta alcun numero, l'oggetto **f** non va in stato di errore
   d) Non è vero che, nel caso in cui tale istruzione sia eseguita con successo, ossia venga memorizzato correttamente un numero intero letto da **f** nella variabile **i**, allora l'*ifstream* è sempre svuotato di tutto il suo contenuto

8. Quali delle seguenti affermazioni sono vere?
   a) In C non esiste il passaggio di un oggetto per riferimento, ma si può emularlo passando l'indirizzo dell'oggetto stesso
   b) La funzione **printf** della libreria standard del C determina automaticamente il formato con cui stampare ciascuno oggetto passato come argomento (dopo la stringa *format*) in base al tipo dell'oggetto stesso
   c) In C, per scrivere in una variabile un valore letto da *stdin* usando la funzione **scanf**, è necessario passare alla funzione sia l'indirizzo che il tipo della variabile in cui scrivere il valore letto
   d) Nella funzione **scanf** della libreria standard del C, sbagliare il tipo o l'indirizzo della variabile in cui si desidera scrivere il valore letto da *stdin* può causare corruzione della memoria o terminazione forzata del programma

9. La memoria occupata da un oggetto allocato dinamicamente all'interno di una funzione
   a) è pre-allocata all'inizio dell'esecuzione del programma;
   b) è allocata quando l'istruzione di allocazione viene eseguita;
   c) è liberata quando la funzione termina;
   d) se non esplicitamente deallocata, è liberata solo al termine del programma.

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

```
main()
{
    bool finito = false ;
    for( int i = 0 ; ! finito ; ) {
          cout<<i<<" " ;
          if (i == 2)
                finito = true ;
          cout<<i<<" " ;
          i++ ;
    }
}
```

11. **(6 pt**) Scrivere una funzione che ha come parametro di ingresso una stringa e ritorna una nuova stringa (precedentemente non esistente in memoria) che contiene gli stessi caratteri della stringa in ingresso, ma in cui ciascun carattere di indice dispari (partendo da 0) è stato scambiato col precedente. Ad esempio: presa in ingresso la stringa *"abcde"*, la funzione deve ritornare la stringa *"badce"*.

Programmazione I
Prova scritta - 22 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					**3**	**-0,5**
4					**3**	**-0,5**
5						
6						
7						
8						
9						

Risposta alla domanda 10 (3 pt):

Risposta alla domanda 11 (6 pt):

Programmazione I
Prova scritta - 22 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					**3**	**-0,5**
4					**3**	**-0,5**
5						
6						
7						
8						
9						

Risposta alla domanda 10 (3 pt):

Risposta alla domanda 11 (6 pt):