Programmazione I
Prova scritta - 7 luglio 2011 - 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, -.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

2. **(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 ;
}
```
a) Se eseguito, il programma stampa **4 3**
b) Se eseguito, il programma stampa **4 4**
c) Nessuna delle altre risposte è vera
d) Il programma causa un errore a tempo di compilazione

3. **(4 pt, -0.5)** Dato il seguente programma:
```
int *a, num_elem ;
void distruggi() { delete [] a; num_elem = 0 ; }
void inserisci(int n) {a[num_elem] = n ; num_elem++ ;}
void stampa() { for (int i = 0 ; i < num_elem ; i++) cout<<a[i]<<" " ; }
```

```
main(){
    a = new int[10] ;
    inserisci(1) ; stampa() ;
    distruggi() ;
    int *b = new int[5] ; b[0] = 3 ;
    inserisci(4) ; stampa() ; cout<<b[0]<<endl ;
}
```

a) il programma stampa **1 4 3**
b) nessuna delle altre risposte è vera
c) il programma contiene un errore di *memory leak*
d) il programma contiene un errore di gestione della memoria diverso dal *memory leak*

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 (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**

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**

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
d) alla riga 11 la variabile **b** contiene ancora una stringa correttamente rappresentata, anche se diversa da *"paolo"*

6. 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

7. Dato il seguente programma:

```
const int N = 1000 ;
struct ss { int b; char a[N] ; } ;
void fun(ss d) { cout<<d.b ; d.a[N/2] = 1 ; d.b = 3 ; }
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 l'intero contenuto del parametro attuale **c** è copiato nel parametro formale **d** (quindi in particolare l'array contenuto nel campo **a** del parametro attuale è riversato nel campo **a** del parametro formale)
   b) In quanto al campo **a** del parametro attuale **c** passato alla funzione **fun**, solo l'indirizzo dell'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: `13 13 2`

8. Dato un programma scritto in linguaggio C/C++
   a) Il record di attivazione di ogni funzione contiene (oltre ad altre informazioni) le variabili locali della funzione (se presenti)
   b) Il record di attivazione di ogni funzione contiene (oltre ad altre informazioni) i parametri formali della funzione (se presenti)
   c) Quando una funzione ne invoca un'altra, viene prima deallocato il record di attivazione della funzione chiamante e quindi creato quello della funzione chiamata
   d) Quando inizia l'esecuzione del programma vengono allocati in memoria i record di attivazione di tutte le funzioni in esso definite

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, supponendo che l'utente immetta **`duecento`** da *stdin*?

```
main()
{
      char stringa[10] = "esatto";
      for (int i = 0 ; i < 3 ; i++)
            cin>>stringa[i] ;
      cout<<stringa<<endl ;
}
```

10. **(6 pt)** Scrivere una funzione che prenda in ingresso due stringhe e stampi la posizione di tutte le occorrenze della prima stringa all'interno della seconda. Ad esempio, se si passa in ingresso la coppia (*"ab"*, *"abcdabd"*), la funzione deve stampare: **0 4**

Programmazione I
Prova scritta - 5 luglio 2011

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 la copia da tenere 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					**4**	**-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 - 5 luglio 2011

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 la copia da tenere 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					**4**	**-0,5**
4					**3**	**-0,5**
5						
6						
7						
8						

Risposta alla domanda 9 (4 pt):

Risposta alla domanda 10 (6 pt):