Programmazione I
Prova scritta - 13 gennaio 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)** Dato il seguente programma e facendo attenzione a cosa accade nel caso in cui l'oggetto **cin** vada in stato di errore:

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

se l'utente inserisce da *stdin* i caratteri `2` e `3`, separati da uno spazio, quindi preme il tasto Invio, ed infine preme CTRL+D (su sistema UNIX)

   a) il programma stampa su *stdout* `2, 3,`
   b) il programma stampa su *stdout* `2, 3, 10,`
   c) il programma stampa su *stdout* `2, 3, 3,`
   d) nessuna delle altre risposte è corretta

2. **(3, -.5)** Il seguente frammento di codice:

```
char c; cin>>c;
if (static_cast<int>(c) > 60)
 cout<<"Maggiore" ;
```

   a) stampa o meno **Maggiore** a seconda del valore della costante carattere **'c'**
   b) stampa **Maggiore** per ogni valore maggiore di 60 memorizzato in **c**
   c) contiene un errore di gestione della memoria
   d) nessuna delle altre risposte è corretta

3. **(2, -.5)** Il seguente frammento di codice

```
int i = 5;
for( ; i>0 ; ) {
  if (i == 2)
      break;
  cout<<i<<" ";
  i--;
}
```

a) stampa gli interi tra 5 e 3 (estremi inclusi);
b) stampa gli interi tra 5 e 1 tranne il 2;
c) stampa gli interi da 5 in su;
d) non stampa alcunché poiché la `cout<<i<<" ";` interna al ciclo non viene mai eseguita.

4. **(4, -.5)** Si consideri l'invocazione della seguente funzione, e si supponga che l'array **a** abbia dimensione uguale al valore dell'argomento **size**, e che **v**>0 e **size**>0:

```
int fun(int a[], int v, int size)
{
     int count = 0, i = 0 ;

     while (i < size)
          if (a[i++]%v == 0)
               break;
          else
               count++ ;
     return count ;
}
```

a) se l'*array* **a** contiene almeno un elemento multiplo di **v** la funzione ritorna l'indice del primo di tali elementi, altrimenti ritorna il valore **size**;
b) la funzione ritorna il numero di elementi multipli di **v** contenuti all'interno dell'array **a**;
c) se l'array non contiene nessun multiplo di **v**, il ciclo **while** è infinito;
d) la funzione contiene un errore di gestione della memoria

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. Due algoritmi equivalenti:
   a) Se implementati danno luogo allo stesso programma
   b) Prevedono gli stessi passi
   c) A parità di dati di ingresso non possono che portare a risultati uguali
   d) A parità di dati in ingresso possono avere tempi di esecuzione diversi

6. Data una stringa in C/C++, rappresentata mediante il seguente array
   `char stringa[20] = "AB" ;`
   a) per ottenere la stringa vuota è necessario assegnare '**\0**' a tutti gli elementi dell'*array*;
   b) la lunghezza della stringa è pari a quella dell'*array* utilizzato per rappresentarla;
   c) purché non si superino le dimensioni dell'*array* meno uno, si può variare a piacimento il contenuto e la lunghezza della stringa (intesa in senso astratto come sequenza di caratteri);
   d) per trasformare la stringa in "ABC", è sufficiente assegnare i valori 'C' e '\0' agli elementi di indice, rispettivamente 2 e 3 dell'*array* (supponendo che gli indici siano numerati a partire da 0).
7. Dato il seguente programma e facendo attenzione ai modi in cui un identificatore può essere nascosto:

```
1: double a = 3.1 ;
2:
3: void fun(float a, int b)
4: {
5:    a += b ;
6: }
7: main()
8: {
9:    fun(static_cast<int>(a), 3) ;
10:   cout<<a<<endl ;
11:}
```

a) la variabile **a** definita alla riga 1 ha scope relativo a tutto il programma;
b) la variabile **a** definita alla riga 1 ha tempo di vita relativo a tutto il programma;
c) prima dell'assegnamento, la variabile **a** riferita alla riga 5 ha un valore diverso da 3.1;
d) l'assegnamento alla riga 5 fa aumentare il valore della variabile **a** definita alla riga 1.

8. Definita una *parola* come una sequenza di caratteri non separati da spazi, il seguente programma:

```
struct parola {char *stringa ; int lun ;} ;

main()
{
   parola v[2] ;
   for (int i = 0 ; i < 2 ; i++) {
         cin>>v[i].stringa ;
         v[i].lun = strlen(v[i].stringa) ;
         cout<<v[i].stringa<<" "<<v[i].lun<<endl ;
   }
}
```

a) legge correttamente da *stdin* due parole e stampa su *stdout* ciascuna delle due parole seguita dalla rispettiva lunghezza
b) contiene un errore di gestione della memoria
c) può comportarsi in modo non prevedibile
d) è sempre terminato forzatamente prima di aver stampato le due parole

9. 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) Per deallocare un record di attivazione è necessario reinizializzare il contenuto delle celle di memoria precedentemente occupate dal record stesso
   c) Il tempo necessario per inizializzare il contenuto di un record di attivazione aumenta all'aumentare del numero e delle dimensioni dei parametri formali
   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

10. 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 ;
}
```

a) La modalità con cui è passata la variabile **c** permette a **fun** di modificarla
b) Il passaggio di **c** per riferimento ha come costo la copia nel parametro formale **d** del contenuto di **c** stesso
c) Il programma stampa **`10 2 3`**
d) L'esecuzione di **fun(c)** ha effetti collaterali sulla porzione della funzione **main** che segue l'invocazione della funzione **fun** stessa

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 stampa su *stdout* il seguente programma, supponendo che l'utente immetta **`duemila`** su *stdin*?

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

12. **(5pt.) Dato un file testo di nome *testo.txt*, si scriva un frammento di codice che esegua le seguenti operazioni:**
    - apra il file in lettura
    - legga da *stdin* due caratteri **c1** e **c2**
    - stampi su *stdout* il numero di volte in cui la sequenza di caratteri **c1 ? ? c2** appare nel file *testo.txt*, dove **?** sta per *qualsiasi carattere*.

Programmazione I
Prova scritta - 13 gennaio 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 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					**2**	**-0,5**
4					**4**	**-0,5**
5						
6						
7						
8						
9						
10						

Risposta alla domanda 11 (3 pt):

Risposta alla domanda 12 (5 pt):

Programmazione I
Prova scritta - 13 gennaio 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. 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					**2**	**-0,5**
4					**4**	**-0,5**
5						
6						
7						
8						
9						
10						

Risposta alla domanda 11 (3 pt):

Risposta alla domanda 12 (5 pt):