Programmazione I
Prova scritta - 21 febbraio 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. **(2, -.5)** Dato un file sorgente C/C++ contenente la seguente direttiva:

```
#define COSTANTE 15
```

a) tale direttiva è perfettamente equivalente all'istruzione

```
const int COSTANTE = 15 ;
```

b) il preprocessore sostituisce testualmente ogni occorrenza della parola **COSTANTE** con la sequenza di due caratteri `15`

c) la parola **COSTANTE** è ancora presente nel file prodotto dal preprocessore e preso in ingresso per la traduzione

d) la presenza di tale direttiva e della seguente istruzione

```
int COSTANTE ;
```

all'interno di una qualche funzione definita in un punto successivo del file (successivo rispetto a quello in cui è presente la direttiva) non comporta errori di compilazione

2. **(3, -0.5)** Dato il seguente programma:

```
struct ts {char s[10] ; int n ;} ;

main(){
      ts a ;
      strcpy(a.s, "prova") ; // memorizzazione della stringa "prova" nel
                             // campo a.s
      a.n = 10 ;
      ts b ; b = a ;
      cout<<b.s<<" "<<b.n ;
}
```

a) l'istruzione di assegnamento *b=a ;* da luogo ad un errore a tempo di compilazione

b) il programma contiene un errore di gestione della memoria

c) il programma stampa `prova 10`

d) nessuna delle altre risposte è vera

3. **(2, -.5)** Il seguente programma:

```
int a = 11, b = 20 ;
int &fun(bool f)
{
 if (f) return a ;
 else return b ;
}
main()
{
 int &c = fun(true) ; a++ ; b-- ;
 cout<<c;
}
```

a) Stampa 11
b) Stampa 12
c) Stampa 19
d) Nessuna delle altre risposte è corretta

4. **(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

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. Due algoritmi equivalenti:
   a) Devono essere codificati entrambi utilizzando lo stesso linguaggio
   b) Hanno sempre lo stesso costo computazionale
   c) Possono utilizzare quantità di memoria differenti durante la loro esecuzione
   d) Possono essere basati su strutture dati diverse

7. Dato il seguente frammento di programma:

```
enum stato {pieno, vuoto} ;
stato v[3] = {pieno, pieno, vuoto} ;
bool fun(int i)
{
      if ( ( i < 0 && i >= 3 )  ||  v[i] == pieno )
            return false ;
      return true ;
}
```

   a) Se invocata passando il valore 2, la funzione *fun* ritorna *true*
   b) Il frammento di programma contiene un errore logico
   c) Il frammento di programma contiene un errore di accesso alla memoria
   d) Nessuna delle altre risposte è vera

8. Dato il seguente programma:

```
int fun(int b[], int n)
{
 int s = 0 ;
 for (int i = 0 ; i < n ; i++) {  s += b[i] ; }
 delete [] b ;
 return s ;
}

main()
{
 int *a = new int[2] ; a[0] = 3 ; a[1] = 4 ;

 cout<<fun(a, 2)<<" " ;
 for (int i = 0 ; i < 2 ; i++)
      cout<<a[i]<<" " ;
}
```

   a) il programma stampa **7 3 4**
   b) nella funzione **fun**, all'interno del ciclo **for** non si modifica l'array dinamico allocato con l'operatore **new** all'inizio della funzione **main**
   c) quando l'esecuzione arriva all'inizio del ciclo **for** nella funzione **main**, il puntatore **a** non punta ad un oggetto correttamente allocato in memoria
   d) il programma contiene un errore di gestione della memoria

9. Dato l'array

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

   le istruzioni:

```
s[2] = 't' ; s[3] = s[2] ; s[4] = 'o' ; s[8] = '+' ;
```

   a) contengono uno o più errori di accesso alla memoria ;
   b) trasformano la stringa da “Soprano” a “Sottono +”;
   c) trasformano la stringa da “Soprano” a “Sottono”;
   d) non modificano la lunghezza della stringa.

10. Nel seguente frammento di codice:

```
1: funzione1()
2: {
3:   int bz=5;
4:   float ak=3.2;
5:   int ps=9;
6:   {
7:     int ps;
8:     int cj;
9:     ps=bz+5;
10:   }
11: }
```

a) le variabili **bz**, **ak**, **ps** definite nelle righe 3-5 hanno tempo di vita e scope relativo a tutto il blocco della **`funzione1()`**

b) le variabili **bz**, **ak**, **ps** definite nelle righe 3-5 hanno tempo di vita relativo a tutto il blocco della **`funzione1()`**

c) le variabili **bz**, **ak**, **cj** hanno scope relativo a tutto il blocco della **`funzione1()`** eccetto il blocco interno

d) le variabili **bz**, **ak** hanno scopo relativo a tutto il blocco della **`funzione1()`**, mentre la variabile **ps** definita riga 5 ha scope relativo a tutto il il blocco della **`funzione1()`** eccetto il blocco interno

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

```
#include <iostream>
#include <ctype.h>
#include <string.h>
using namespace std ;

int fun(char *s);

int main()
{
     int f = 0;
     char stringa[] = "EsameInfo";
     cout<<stringa<<", "<<f<<endl;
     f = fun(stringa);
     cout<<stringa<<", "<<f<<endl;
     for(int i = strlen(stringa) - 1 ; i >= 0 ;i--)
         cout<<stringa[i];
     cout<<endl ;
     return 0;
}
```

```
int fun(char *s)
{
    int i = 0, n = 0;
    while (s[i] != '\0'){
     if (isupper(s[i])){
         s[i] = tolower(s[i]);
         ++n;
     } else
         cout<<s[i];
     ++i;
    }
    cout<<endl ;
    return n;
}
```

12. **(3 pt)** Scrivere una funzione che prenda un ingresso una stringa e, all'interno della stringa, trasformi ciascuna lettera maiuscola in minuscola, e viceversa. Si possono utilizzare, tra le funzioni di libreria delle stringhe, solo le funzioni *to_lower* e *to_upper* (che prendono in ingresso un carattere e, se si tratta di una lettera maiuscola/minuscola ritornano la corrispondente lettera minuscola/maiuscola).

Programmazione I
Prova scritta - 21 febbraio 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					**2**	**-0,5**
2					**3**	**-0,5**
3					**2**	**-0,5**
4					**2**	**-0,5**
5					**3**	**-0,5**
6						
7						
8						
9						
10						

Risposta alla domanda 11 (5 pt):

Risposta alla domanda 12 (3 pt):

Programmazione I
Prova scritta - 21 febbraio 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					**2**	**-0,5**
2					**3**	**-0,5**
3					**2**	**-0,5**
4					**2**	**-0,5**
5					**3**	**-0,5**
6						
7						
8						
9						
10						

Risposta alla domanda 11 (5 pt):

Risposta alla domanda 12 (3 pt):