Programmazione I / Informatica generale
Prova scritta - 14 gennaio 2009 - 1h30min

NOTA: Nei programmi si trascuri ogni problema legato al tipo ed al valore di ritorno della funzione **main** e si sottintenda la presenza delle direttive
`#include <iostream>  /  #include <fstream>  /  using namespace std ;`
Inoltre, laddove si trovi l'affermazione che un programma o frammento di codice fornisce un certo risultato, è da intendersi che, in accordo alle regole del linguaggio, tale programma o frammento di codice fornisce quel risultato per **qualsiasi esecuzione** su **qualsiasi macchina**. Infine, per fallimento di un programma è da intendersi l'esecuzione di una operazione illegale (quale ad es.: una divisione per 0 o un accesso illegale in memoria) da parte del programma e la sua conseguente terminazione forzata da parte del sistema operativo.

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 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) Se invocata passando il valore 3, il programma fallisce durante l'esecuzione della funzione **fun**

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. **(4, -0.5)** Dato il seguente programma e considerando attentamente possibili problemi di *overflow*:

```
main() {
    int i ; cin>>i ; i = static_cast<unsigned int>(i) ;
    if (i == 0 || i == 1) i = 2 ;
    for (unsigned int j = 5 ; j > 0 ; ) {
          j /= static_cast<unsigned int>(i) ;
          cout<<j<<" " ;
    }
}
```

   a) Esiste almeno un numero negativo tale che, se inserito dall'utente, fa sì che il programma esegua un ciclo infinito
   b) Qualsiasi numero negativo inserisca l'utente, il programma stampa `2 1 0`
   c) Se l'utente inserisce `-2` il programma stampa `-2`
   d) Nessuna delle altre risposte è corretta

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**

4. Indicare quali delle seguenti affermazioni sono vere
   a) Dato un programma scritto in un linguaggio che può essere sia compilato che interpretato, l'esecuzione di tale programma da parte di un interprete è tipicamente più lenta dell'esecuzione della versione compilata del programma
   b) Il linguaggio macchina permette di scrivere programmi portabili tra diverse architetture
   c) Il linguaggio macchina non è un linguaggio di alto livello
   d) Un programma in linguaggio macchina è una sequenza di byte non interpretabili come sequenza di caratteri

5. Dato il seguente programma:

```
const int N = 2000 ;
struct ss { int b; char a[2000] ; } ;
void fun(ss d) { cout<<d.b ; d.a[N/2] = 1 ; d.b = 3 ; }
main()
{
 ss c ; c.b = 12 ; 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: `12 12 2`

6. Dato il seguente programma

```
int fun(int &g) { return 2 * (--g); }

int g = 2 ;
main() {
  char n = static_cast<int>(3.4) * fun(g) ;
  cout<<static_cast<int>(g * n) ;
}
```

a) Se eseguito, il programma stampa **6**
b) Nell'ultima istruzione del **main**, la conversione **static_cast<int>(g * n)** non causa perdita di informazione
c) Nessuna delle altre risposte è vera
d) Dopo l'invocazione della funzione **fun** nella prima istruzione del **main**, la variabile globale **g** ha un valore diverso da quello iniziale

7. La seguente istruzione:

```
cout<<static_cast<char>('c' - 'a' + '3') ;
```

a) non può causare problemi di *overflow*
b) immette il codice di un carattere stampabile sullo *stream* di uscita standard (**cout**)
c) stampa **5**
d) nessuna delle altre risposte è corretta

8. Contrassegnare la/le affermazione/i corretta/e:
a) A differenza di un oggetto di tipo **int**, un oggetto di tipo **long long int** non è soggetto a problemi di *overflow*
b) Un oggetto di tipo **long long int** può occupare in memoria uno spazio maggiore o uguale di un oggetto di tipo **int**
c) Un oggetto di tipo **float** ha per definizione una precisione (numero di cifre significative dei numeri che può rappresentare) maggiore di un **long long int**
d) Un oggetto di tipo **double** può occupare meno memoria di un oggetto di tipo **float**

PARTE 3 – DOMANDE APERTE –

- **Una risposta esatta fa acquisire il punteggio positivo riportato a fianco della domanda**
- **Una risposta errata** può **causare una penalità che dipende dalla gravità dell'errore, ed al più uguale al punteggio negativo riportato a fianco della domanda**
- **Una risposta lasciata in bianco viene calcolata: 0**

9. **(5 pt)** Cosa viene stampato su *stdout* dal seguente programma?

```
int fun(const char s[])
{
   int j = strlen(s) - 1 ;
   int c = 0 ;
   for (int i = j ; i >= 0 ; i--) {
      if (s[i] != ' ')
           c++ ;
      else {
           if (c == 3)
                return i ;
           c = 0 ;
      }
   }
   return -1 ;
}
```

```
main() {
    const char s[] = "Nel mezzo del cammin di nostra  vita" ;
    for (int g = fun(s) ; s[g] != 'n' ; g++)
          cout<<s[g+1] ;
}
```

10. **(5 pt.)** Scrivere una funzione che prenda in ingresso un vettore di *N* interi e lo trasformi in un vettore contenente prima tutti gli elementi di indice dispari e poi tutti quelli di indice pari del vettore iniziale. Supporre che gli indici partano da 0. Esempio: preso in ingresso il vettore {2, 1, 5, 8, 3}, la funzione deve trasformarlo in {1, 8, 2, 5, 3}.

Programmazione I / Informatica generale
Prova scritta - 14 gennaio 2009

Nome: _____ Cognome: _____

Matricola: _____ Corso di Laurea: _____

Indicare le risposte corrette apponendo una croce nella casella corrispondente. Per superare la prova bisogna aver raggiunto almeno 10 punti nelle domande a risposta singola/multipla, ed almeno 15 complessivamente. Avete due copie di questa pagina, usatene una per calcolare il voto da sole/soli durante la correzione e decidere se consegnare.

	Risposte				Punti/ Penalità	
	A	**B**	**C**	**D**		
1					**3**	**-0.5**
2					**3**	**-0.5**
3					**4**	**-0.5**
4						
5						
6						
7						
8						

Risposta alla domanda 9 (5 pt):

Risposta alla domanda 10 (5 pt):

Programmazione I / Informatica generale
Prova scritta - 14 gennaio 2009

Nome: _____ Cognome: _____

Matricola: _____ Corso di Laurea: _____

Indicare le risposte corrette apponendo una croce nella casella corrispondente. Per superare la prova bisogna aver raggiunto almeno 10 punti nelle domande a risposta singola/multipla, ed almeno 15 complessivamente. Avete due copie di questa pagina, usatene una per calcolare il voto da sole/soli durante la correzione e decidere se consegnare.

	Risposte				Punti/ Penalità	
	A	**B**	**C**	**D**		
1					**3**	**-0.5**
2					**3**	**-0.5**
3					**4**	**-0.5**
4						
5						
6						
7						
8						

Risposta alla domanda 9 (5 pt):

Risposta alla domanda 10 (5 pt):