Programmazione I
Prova scritta - 25 febbraio 2015 - 1h20min

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 valutata 0**

1. **(3, -0.5)** Dato il seguente programma e supponendo che l'operatore >> non modifichi il valore della variabile se l'oggetto **cin** è oppure entra in stato di errore:

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

se l'utente inserisce da *stdin* i caratteri `1` e `3`, separati da uno spazio, quindi preme Invio, ed infine preme CTRL+D (su sistema UNIX)

a) il programma stampa su *stdout* `1, 3,`
b) il programma stampa su *stdout* `1, 3, 10,`
c) il programma stampa su *stdout* `1, 3, 3,`
d) nessuna delle altre risposte è corretta

2. **(3, -0.5)** Supponendo che la matrice `int a[2][5]` sia memorizzata a partire dall'indirizzo 100 e che gli oggetti di tipo `int` occupino 4 byte, l'indirizzo dell'elemento `a[1][2]` è

a) 100 + 0*4*5 + 2*4 = 108
b) 100 + 1*4*2 + 1*4 = 112
c) 100 + 1*4*5+ 2*4 = 128
d) nessuna delle altre risposte è corretta

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

```
void fun(int a[])
{ int n; cin>>n ; for (int i = 0 ; i < n ; i++) cin>>a[i] ;}

main() { int c[3] ; fun(c) ; }
```

a) Il compilatore controlla che la funzione **fun** non modifichi l'array che le viene passato
b) La funzione **fun** modifica solo una copia locale dell'array **c** che le viene passato
c) La variabile **n** ha classe di memorizzazione automatica
d) Nessuna delle altre risposte è corretta

PARTE 2 – (POSSIBILI) RISPOSTE MULTIPLE - Ogni domanda può avere da una a quattro 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) A parità di algoritmo implementato, l'esecuzione di un programma scritto in un linguaggio compilato (e quindi poi compilato) è tipicamente più veloce dell'esecuzione di un programma scritto in un linguaggio interpretato
   b) Il linguaggio macchina permette di scrivere programmi portabili tra diverse architetture di calcolatori
   c) L'esecuzione dell'invocazione di una funzione causa l'aumento delle dimensioni dello *stack*
   d) Un programma in linguaggio macchina è una sequenza di byte da non interpretarsi come sequenza di caratteri

5. 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) In C non esiste il passaggio per riferimento e non c'è nessun modo per emularlo
   c) Dato un parametro formale di tipo puntatore ad un certo tipo **T**, allora, all'atto del passaggio dell'indirizzo di un oggetto di tipo **T** come parametro attuale nella posizione corrispondente a tale parametro formale, non avviene la copia del contenuto dell'oggetto di tipo **T** memorizzato a tale indirizzo
   d) Nessuna delle altre affermazioni è vera

6. Dato il seguente programma:

```
const int N = 100 ; struct ss { int b; char a[N] ; } ;
void fun(ss & d) { cout<<d.a[N/2]<<" "; d.b = 4 ; }
main()
{
   ss c ; c.b = 7 ; c.a[N/2] = '2' ; cout<<c.b<<" " ;
   fun(c) ; cout<<c.b ;
}
```

a) Il passaggio di **c** per riferimento permette a **fun** di modificare **c**
b) Il programma stampa **7 2 4**
c) Il passaggio di **c** per riferimento ha come costo la copia nel parametro formale **d** del contenuto di **c** stesso
d) L'esecuzione di **fun(c)** ha effetti collaterali sulla porzione della funzione **main** che segue l'invocazione della funzione **fun** stessa

7. Facendo attenzione ad eventuali problemi di *overflow* e supponendo che la costante carattere `'w'` contenga un valore maggiore di zero, il seguente frammento di codice:

```
int i; cin>>i;
if (static_cast<unsigned char>(i) > 'w')
 cout<<"Maggiore" ;
```

a) stampa o meno **Maggiore** in base al valore assunto dalla variabile **i**
b) stampa **Maggiore** per ogni valore di **i** maggiore del valore della costante carattere `'w'`
c) può stampare **Maggiore** per qualche valore di **i** minore di zero
d) nessuna delle altre risposte è corretta

8. 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) Se l'indice **i** è compreso tra 0 ed N-1 (estremi inclusi) la funzione ritorna **true**
b) Nessuna delle altre risposte è vera
c) Solo se l'indice **i** non 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
d) La funzione potrebbe leggere al di fuori dell'array **a**

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. **(6 pt)** Descrivere la motivazione per l'uso di *array* dinamici e gli svantaggi dell'uso di *array* dinamici rispetto ad *array* automatici, in non più di 8 righe (non verrà valutata la quantità ma la qualità di quello che si scrive, ed il tentativo di scrivere in modo estremamente fitto per aumentare la quantità, così come il superamento del numero massimo di righe, comporteranno una penalità).

10. **(7 pt)** Scrivere una funzione che prende in ingresso una stringa contenente solo lettere minuscole, e ritorna una nuova stringa contenente solo i caratteri presenti almeno due volte nella stringa in ingresso. Ad esempio, presa in ingresso la stringa *"estense"*, la funzione ritornerebbe la stringa *"es"*. Si ottiene il punteggio massimo se si realizza una soluzione con ordine di costo computazionale minimo, anche se tale soluzione richiede un'occupazione di memoria maggiore di altre.

Programmazione I
Prova scritta - 25 febbraio 2015

Nome: _____ Cognome: _____

Matricola: _____

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 una 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						
5						
6						
7						
8						

Risposta alla domanda 9 (6 pt):

Risposta alla domanda 10 (7 pt):

Programmazione I
Prova scritta - 25 febbraio 2015

Nome: _____ Cognome: _____

Matricola: _____

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. Usate questa copia per calcolare il voto da sole/soli durante la correzione.

	Risposte				**Punti/**	
	A	**B**	**C**	**D**	**Penalità**	
1					**3**	**-0,5**
2					**3**	**-0,5**
3					**2**	**-0,5**
4						
5						
6						
7						
8						

Risposta alla domanda 9 (6 pt):

Risposta alla domanda 10 (7 pt):