Programmazione I
Prova scritta - 13 febbraio 2018 - 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 non si verifichi alcun problema di rappresentazione approssimata dei numeri reali coinvolti

```
int g = 1 ;
int fun() { ++g ; return 7 ; }
main()
{ double l = static_cast<double>(fun()) / 2 ;
  cout<<static_cast<int>(l * g) ; }
```

   a) Se eseguito, il programma stampa 7
   b) Se eseguito, il programma stampa 6
   c) Nessuna delle altre risposte è corretta
   d) Nell'ultima istruzione del **main**, la conversione esplicita causa perdita di informazione

2. **(3, -.5)** Supponendo che *dati.dat* sia un file binario e che il tipo *char* abbia dimensione pari a quella di un byte, il seguente frammento di codice:

```
ifstream f("dati.dat"); ofstream d("secondo_file") ;
char c;
f>>noskipws ; // non saltare alcun codice carattere
while (f>>c)
      d<<c;
```

   a) causa una terminazione forzata a tempo di esecuzione
   b) nessuna delle altre risposte è vera
   c) crea un file di nome *secondo_file* identico al file *dati.dat*
   d) crea un file di nome *secondo_file* il cui contenuto può differire da quello di *dati.dat*

3. **(2, -.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 contenuto dell'*array* **b** è copiato nell'*array* **a**
   b) Quando la funzione **fun** è invocata, l'indirizzo del (primo elemento) dell'*array* **b** 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

4. **(3, -.5)** Data la seguente stringa rappresentata mediante un array
   ```
   char s[10] = "Sol" ;
   ```
   le istruzioni:
   ```
   s[2] = 't' ; s[3] = s[2] ; s[4] = 'o' ; s[5] = '\0' ;
   ```
   a) contengono uno o più errori di accesso alla memoria
   b) trasformano la stringa da "Sol" a "Sotto", aumentandone la lunghezza
   c) non modificano la lunghezza della stringa, ma non contengono errori di accesso alla memoria
   d) nessuna delle altre risposte è vera

PARTE 2 – (POSSIBILI) RISPOSTE MULTIPLE - Ogni domanda può avere <u>da una a quattro</u> risposte CORRETTE.

- **Ogni risposta esatta viene calcolata: +1**
- **Ogni risposta errata viene calcolata: -.5**
- **Una risposta lasciata in bianco viene calcolata: 0**

5. Data una sequenza di elementi tutti dello stesso tipo
   a) Se le operazioni più frequenti sulla sequenza sono inserimenti ed estrazioni in/dal mezzo di elementi di cui si conosce la posizione in memoria, allora una lista è la struttura dati più conveniente da utilizzare, in termini di costo computazionale, per implementare la sequenza
   b) Se, all'inizio della propria esecuzione, il programma è informato prima del numero effettivo di elementi della sequenza e poi del contenuto di ciascuno di tali elementi, e tale numero di elementi non varierà più durante l'esecuzione programma, allora, in termini di costo computazionale ed occupazione di memoria, la struttura dati più conveniente da utilizzare per implementare la sequenza è un array dinamico
   c) Se l'ordine degli elementi conta, il numero di elementi che saranno inseriti nella sequenza è noto a tempo di scrittura del programma e non avverranno mai estrazioni, allora la struttura dati più conveniente da utilizzare, in termini di semplicità di gestione, per implementare la sequenza è un array dinamico
   d) Un array utilizzato per memorizzare tale sequenza può occupare più memoria di una lista che rappresenti la stessa sequenza, se alcuni degli elementi dell'array non rappresentano alcun elemento della sequenza

6. Dato il seguente programma:

```
 1:int a = 7;
 2:
 3:main()
 4:{
 5:    for (int i = 4 ; i > 2 ; i--) {
 6:       int a = i / 2 ;
 7:       cout<<(a * i)<<" ";
 8:    }
 9:    cout<<a;
10:}
```

   a) il programma stampa **8 3 1**
   b) la variabile **a** definita nella riga 1 non è visibile in tutta la funzione **main**
   c) il programma stampa **8 3 7**
   d) la variabile **i** è visibile anche alla riga 9

7. Assumendo che i valori di tipo **double** siano rappresentati mediante lo standard IEEE 754 in base 2:
   a) Il tipo **double** non può rappresentare tutti i numeri reali compresi nel suo intervallo di rappresentabilità
   b) La precisione del tipo **double** dipende dal numero di cifre riservate alla rappresentazione dell'esponente
   c) Se **a**, **b** e **c** sono tre variabili di tipo double, **a** e **b** contengono due valori positivi, e si esegue l'assegnamento **c = a + b**, senza che la somma **a + b** generi *overflow*, allora, dopo tale assegnamento, la seguente espressione logica può essere lo stesso falsa: **a == c – b**.
   d) Nessuna delle altre risposte è vera

8. Dato un puntatore **int *p**, che punta ad un *array* di **int** allocato dinamicamente in memoria, l'istruzione: `p=0;`
   a) fa sì che **p** non punti più ad alcun *array* in memoria dinamica;
   b) anche se **p** non era l'unico riferimento alla zona di memoria, impedisce di eliminare successivamente l'oggetto dalla memoria;
   c) non elimina l'*array* dalla memoria dinamica;
   d) azzera anche il valore di eventuali altri puntatori che puntano allo stesso *array*.

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 sintassi, semantica e motivazione degli array dinamici, in non più di 10 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 prenda in ingresso un vettore **v** di interi (qualsiasi) ed un valore intero *n*, e rimuova dal vettore **v** tutti gli elementi che non appartengono alla più grande sotto-sequenza di elementi adiacenti con le seguenti caratteristiche: il primo elemento della sotto-sequenza ha indice *n*, tutti gli elementi della sotto-sequenza sono ordinati per valori crescenti. Ad esempio, presi in ingresso il vettore [2, 3, 5, 10, 9, 12] ed il valore 1, tale sotto-sequenza sarebbe [3, 5, 10], per cui la funzione trasformerebbe il vettore in [3, 5, 10]. Si ottiene il punteggio massimo se si implementa un algoritmo che ha complessità lineare e che non utilizza ulteriori vettori di appoggio.

Programmazione I
Prova scritta - 13 febbraio 2018

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

Risposta alla domanda 9 (6 pt):

Risposta alla domanda 10 (7 pt):

Programmazione I
Prova scritta - 13 febbraio 2018

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					**3**	**-0,5**
5						
6						
7						
8						

Risposta alla domanda 9 (6 pt):

Risposta alla domanda 10 (7 pt):