Programmazione I
Prova scritta - 27 gennaio 2020 - 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, -.5)** Tenendo presente eventuali problemi di conversione (il tipo **unsigned int** è gerarchicamente superiore al tipo **int**, e qualsiasi operazione tra **unsigned int** ha per risultato un **unsigned int**) e di rappresentabilità, quale della seguenti risposte è vera riguardo al seguente frammento di codice?

```
int delta_w; unsigned int wsum;
cin>>delta_w>>wsum;
if (wsum - delta_w >= 0)cout<<"maggiore"<<endl ;
```

a) Non stampa nulla se **`delta_w`** è positivo e strettamente maggiore di **`wsum`**
b) Stampa **`maggiore`** se **`delta_w`** è negativo e **`wsum`** contiene un valore negativo
c) Stampa **`maggiore`** qualsiasi sia la coppia di valori interi inserita dall'utente
d) Nessuna delle altre risposte è vera

2. **(3, -.5)** Assumendo che i valori di tipo **double** siano memorizzati in accordo allo standard IEEE 754 (in base 2):

a) Il tipo **double** 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 è vera: **a == c – b**.
d) Nessuna delle altre risposte è vera

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

4. **(2, -.5)** Date le istruzioni **`cout<<'4'<<'1'<<endl;`** e **`cout<<41<<endl;`** e supponendo che l'operatore di uscita sia configurato per stampare i numeri interi in base 10.
a) Tali istruzioni immettono sullo *stdout* la stessa sequenza di codici carattere dell'istruzione **`cout<<4<<1<<endl ;`**
b) Nessuna delle altre risposte è vera
c) La seconda istruzione immette su *stdout* un numero minore di byte rispetto alla prima
d) La seconda istruzione immette su *stdout* un numero maggiore di byte rispetto alla prima

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: -.5**
- **Una risposta lasciata in bianco viene calcolata: 0**

5. Data una sequenza di *N* valori interi positivi
a) La struttura dati più efficiente, in termini di tempo di lettura di tutti i valori, in cui si potrebbero memorizzare tali valori è un *array* di *N* elementi
b) La struttura dati più efficiente, in termini di occupazione di memoria, in cui si potrebbero memorizzare tali valori è una lista semplice di *N* elementi
c) Aggiungere un elemento nel mezzo della sequenza ha costo *O(N)* se la sequenza è implementata mediante un array
d) Se la sequenza è memorizzata in una lista semplice e senza memorizzare in alcuna altra variabile aggiuntiva il numero di elementi della sequenza, allora sono necessarie più di *O(1)* operazioni per calcolare tale numero di elementi

6. Dato il seguente programma

```
main() {
   int i = 0 ;
   cin>>i ;
   switch(i) {
   case 'b':
        cout<<"Primo" ;
   case 'c':
        cout<<"Secondo" ;
        break ;
   default:
        cout<<"Errore" ;
   }
}
```

a) Se l'utente immette il codice numerico usato per rappresentare il carattere **`c`**, stampa **`Secondo`**
b) Se l'utente immette il codice numerico usato per rappresentare il carattere **`b`**, stampa **`Primo`**
c) Se l'utente immette il carattere **`b`** stampa **`Errore`**
d) Se l'utente immette il codice numerico usato per rappresentare il carattere **`b`**, stampa **`Errore`**

7. Dato il seguente programma

```
bool fun(int &a) {
if (a > 10) { ++a ; return true ; } return false ; }
main(){
 int m = 10 ; if (m < 30 && fun(m)) cout<<(m*2) ; else cout<<m ; }
```

   a) Se eseguito, il programma stampa 10
   b) Se eseguito, il programma stampa 20
   c) Se eseguito, il programma stampa 22
   d) Il valore della variabile **m** non è modificato in conseguenza della invocazione della funzione **fun** all'interno del **main**

8. Dato il seguente programma:

```
const int N = 10 ; struct ss { char a[N] ; } ;
struct ss fun(ss & e) { e.a[0] = 'z' ;  return e ;}
main()
{
   ss c, d ; c.a[0] = 'y' ;
   d = fun(c) ; d.a[0] = 'k' ; cout<<c.a[0] ;
}
```

   a) Il programma stampa **k**
   b) Il programma stampa **z**
   c) Il ritorno di **e** nella funzione **fun** comporta la copia del contenuto del parametro attuale a cui si riferisce il parametro formale **e**
   d) Quando **fun** è eseguita, il parametro formale **e** diventa un sinonimo per la variabile locale **c**

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. **(6 pt)** Scrivere una funzione che prenda in ingresso un vettore **v** di interi (qualsiasi) ed un valore intero *n*, e rimuova dal vettore **v** la 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, 1, 5, 10, 9, 12] ed il valore 1, la funzione trasforma il vettore in [2, 9, 12]. Si ottiene il punteggio massimo se si implementa un algoritmo di complessità lineare e che non utilizza ulteriori vettori di appoggio.

Programmazione I
Prova scritta – 27 gennaio 2020

Nome: _____ Cognome: _____

Matricola: _____

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 18 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					**2**	**-0,5**
5						
6						
7						
8						

Risposta alla domanda 9 (6 pt):

Risposta alla domanda 10 (6 pt):

Prova Scritta Programmazione I

Potete utilizzare questa pagina, che non dovete consegnare, per scrivere le vostre risposte in un formato che ne semplificherà il controllo durante la correzione pubblica.

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 18 complessivamente. Usate questa copia 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					**2**	**-0,5**
5						
6						
7						
8						

Risposta alla domanda 9 (6 pt):

Risposta alla domanda 10 (6 pt):