

Programmazione I

Prova scritta - 8 giugno 2017 - 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) Dato il seguente programma e facendo attenzione al modo in cui i numeri sono scritti nel file:

```
main()
{
    ofstream f("nome.txt") ;
    f<<1<<2<<endl ; f<<3 ; f.close() ;
    ifstream f2("nome.txt") ;
    int i ;
    while(f2>>i)
        cout<<i<<" , " ;
}
```

- a) al termine del programma *nome.txt* è un file binario contenente i numeri 1, 2 e 3
- b) al termine del programma *nome.txt* è un file di testo contenente i numeri 1, 2 e 3
- c) il programma stampa almeno 1, 2,
- d) nessuna delle altre risposte è corretta

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

```
int *a, num_elem ;
void distruggi() { delete [] a; num_elem = 0 ; }
void inserisci(int n) { a[num_elem] = n ; num_elem++ ; }
void stampa() { for (int i = 0 ; i < num_elem ; i++) cout<<a[i]<<" " ; }

main(){
    a = new int[10] ; inserisci(3) ; stampa() ; distruggi() ;
    a = new int[5] ; inserisci(2) ; stampa() ;
}
```

- a) il programma stampa 3 2
- b) nessuna delle altre risposte è vera
- c) il programma potrebbe essere terminato forzatamente all'atto della seconda allocazione dinamica della memoria, a causa di un errore di gestione della memoria
- d) il programma contiene un errore di gestione della memoria

3. (3,-.5) Se si esegue il seguente programma:

```
main() { char str[20], c; int i = 0 ;  
    while(cin>>c) str[i++] = c ;  
    str[i] = '\0'; cout<<str ;}
```

e l'utente inserisce sullo *stdin* una sequenza, di lunghezza al più 10, di caratteri non separati da spazi, dopodiché segnala la fine dell'input con, ad esempio, Ctrl+D da terminale Unix, allora:

- a) il programma stampa la sequenza di caratteri inserita dall'utente
- b) nessuna delle altre risposte è vera
- c) il programma non stampa nulla
- d) il programma stampa la sequenza di caratteri inserita dall'utente, seguita dal carattere speciale di codice 0

4. (2,-.5) Il seguente programma:

```
int a = 10, b = 20 ;  
int &fun(bool f) { if (f) return a ; else return b ; }  
main() { int &c = fun(false) ; cout<<c; }
```

- a) Stampa 10
- b) Stampa 20
- c) Contiene un errore di gestione della memoria
- 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: -.5
- Una risposta lasciata in bianco viene calcolata: 0

5. Dato un programma scritto in linguaggio C/C++

a) Per deallocare un record di attivazione è necessario reinizializzare il contenuto delle celle di memoria precedentemente occupate dal record stesso

b) Il tempo necessario per inizializzare il contenuto di un record di attivazione aumenta all'aumentare del numero e delle dimensioni dei parametri formali

c) Il record di attivazione di ogni funzione contiene (oltre ad altre informazioni) le variabili globali a cui accede la funzione

d) Il tempo necessario per inizializzare il contenuto di un record di attivazione aumenta all'aumentare del numero e delle dimensioni delle variabili locali inizializzate

6. Dato il seguente programma e supponendo che non sorga alcun problema di approssimazione dovuto alla precisione limitata del tipo **double**:

```
1: double a = 5.2 ;  
2: void fun(int a, int b) { a -= b ; }  
3: main() { fun(a, 4) ; cout<<a<<endl ; }
```

a) il programma stampa 5.2

b) la variabile **a** definita alla riga 1 non ha tempo di vita pari all'intero programma;

c) prima del decremento, la variabile **a** riferita alla riga 2 ha un valore diverso da 5.2;

d) la variabile **a** definita alla riga 1 è visibile alla riga 2

7. Dato il seguente programma e supponendo che il valore massimo per un **char** sia 127:

```
int fun(int a) { return a ; } ;  
int main() {  
    char i = numeric_limits<char>::max() ; double h = 2.8 ;
```

```
const double g = i + fun(h) ; cout<<g ; }
```

- a) Il programma stampa **129**
- b) Il programma stampa un numero diverso da **129**
- c) La somma effettuata nella seconda riga della funzione **main** è di tipo **int**
- d) C'è perdita di informazione nel passaggio del parametro attuale nella invocazione della funzione **fun** nella seconda riga della funzione **main**

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) In non più di 10 righe, descrivere il significato di classe di memorizzazione statica, automatica e dinamica (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 **S**, che si assume contenere la rappresentazione in base 2 di un numero **N**, e ritorna tale numero **N**. Ad esempio, se la stringa **S** è "101", la funzione ritorna 5. La lunghezza massima consentita per la stringa **S** è 10.

Programmazione I

Prova scritta - 20 febbraio 2017

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					3	-0,5
4					2	-0,5
5						
6						
7						
8						

Risposta alla domanda 9 (6 pt):

Risposta alla domanda 10 (7 pt):

Programmazione I

Prova scritta - 20 febbraio 2017

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

Risposta alla domanda 9 (6 pt):

Risposta alla domanda 10 (7 pt):