

Programmazione I

Prova scritta - 10 gennaio 2019 - 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) Il seguente programma:

```
void leggi(char a[], int n){ ifstream f("nome_file");
    for (int i = 0 ; i < n ; i++)
        if (! (f>>a[i]) ) break ;
void scrivi(const char a[], int n){ ofstream f("nome_file");
    f.write(a, sizeof(char) * n); }
main() {
    char b[] = {'d', 'b', 'c'}, c[] = {'d', 'd', 'd'};
    scrivi(b, 3) ; leggi(c, 3) ;
    for (int i = 0 ; i < 3 ; i++) cout<<c[i] ;
}
```

- a) crea un file di testo contenente i caratteri **dbc**
- b) stampa qualcosa di diverso da **dbc**
- c) non memorizza correttamente in **c** i caratteri contenuti in **b** mediante la funzione **scrivi**, perché **b** è stato precedentemente memorizzato in forma binaria nel file **nome**
- d) nessuna delle altre risposte è vera

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[2] ; inserisci(2) ; inserisci(5) ; stampa() ;
}
```

- a) il programma non contiene alcun errore di gestione della memoria e stampa 3 2 5
- b) nessuna delle altre risposte è vera
- c) il programma contiene un errore che causa *memory leak*
- d) il programma contiene un errore che causa l'accesso a zone di memoria non allocate

3. (3, -.5) Data la seguente stringa rappresentata mediante un array

```
char s[10] = "Treno" ;
```

l'istruzione:

```
s[3] = '\0' ;
```

- a) contiene uno o più errori di gestione della memoria
- b) trasforma la stringa da "Treno" a "Tre"
- c) non modifica la lunghezza della stringa
- d) nessuna delle altre risposte è vera

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

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

- a) Il tempo necessario per inizializzare il contenuto di un record di attivazione è indipendente dal numero e le dimensioni delle variabili locali inizializzate
- b) Il record di attivazione relativo ad una data funzione non contiene le variabili globali a cui accede la funzione
- c) L'esecuzione di una istruzione di allocazione dinamica della memoria non provoca nessuna variazione delle dimensioni del record di attivazione della funzione contenente tale istruzione
- d) Un record di attivazione relativo ad una data funzione **fun** contiene anche le informazioni necessarie per far riprendere l'esecuzione della funzione chiamante a partire dall'istruzione successiva alla chiamata di **fun** stessa

5. Dato il seguente programma:

```
void inizializza(int a[][3], int N)
{ for(int i = 0 ; i < N ; i++)
  for(int j = 0 ; j < 3 ; j++)
    a[i][j] = 0 ; }
main() { int c[2][3] ; inizializza(c, 2) ; }
```

- a) La matrice *c* è vuota, ossia i suoi elementi non hanno nessun valore, quando viene passata alla funzione *inizializza*, che poi provvederà a dare un valore a ciascun elemento
- b) In quanto all'accesso agli elementi della matrice *a* all'interno della funzione *inizializza*, conoscere il numero di colonne della matrice è necessario per conoscere l'indirizzo in memoria di ciascun elemento della matrice
- c) In quanto all'accesso agli elementi della matrice *a* all'interno della funzione *inizializza*, conoscere il numero di righe della matrice è necessario per conoscere l'indirizzo in memoria di ciascun elemento della matrice
- d) Dopo l'invocazione della funzione *inizializza*, tutti gli elementi della matrice *c* contengono il valore 0

6. La seguente istruzione:

```
cout<<static_cast<char>('3' - ('d' - 'b')) ;
```

- a) stampa 1
- b) può causare problemi di *overflow*
- c) immette il codice di un carattere stampabile sullo *stream* di uscita standard (**cout**)
- d) nessuna delle altre risposte è corretta

7. Dato il seguente programma:

```
int g = 1 ;
bool fun() { ++g ; return true ;}
main() {
    const int m = 6 ;
    if ((m % 2 == 0) && fun()) cout<<-g ;
    else cout<<g ;
}
```

- a) Se eseguito, il programma stampa **1**
- b) Se eseguito, il programma stampa **-2**
- c) Il valore della variabile globale **g** viene modificato durante l'esecuzione del programma (dopo la sua inizializzazione)
- d) Durante l'esecuzione del programma, è invocata anche la funzione **fun**

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**

8. (7 pt) Descrivere le regole di visibilità degli identificatori in non più di 10 righe più eventuali frammenti di codice (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).

9. (6 pt) Senza utilizzare funzioni o oggetti di libreria per l'output formattato all'interno di stringhe, scrivere una funzione che prenda in ingresso un numero intero e crei e ritorni una nuova stringa contenente la rappresentazione in base 10 del numero, preceduta dal segno del numero. Se, ad esempio, viene passato il numero 34, al funzione ritorna la stringa "+34", mentre se viene passato il numero -278, la funzione ritorna la stringa "-278". Si ottiene il punteggio massimo se si fa l'assunzione che il numero passato possa avere un numero arbitrariamente grande di cifre, e si gestisce correttamente tale assunzione.

Programmazione I
Prova scritta - 10 gennaio 2019

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						
5						
6						
7						

Risposta alla domanda 8 (7 pt):

Risposta alla domanda 9 (6 pt):

Programmazione I
Prova scritta - 10 gennaio 2019

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						
5						
6						
7						

Risposta alla domanda 8 (7 pt):

Risposta alla domanda 9 (6 pt):