

Programmazione I

Prova scritta - 17 febbraio 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) In merito alla correttezza e/o qualità della definizione del parametro formale nella seguente funzione:

```
void fun(struct &f){ cout<<f.c[100]<<endl; }
```

- a) Sarebbe stato necessario aggiungere il qualificatore **const** per correttezza
- b) Sarebbe stato opportuno aggiungere il qualificatore **const** per robustezza
- c) Sarebbe stato più opportuno effettuare un passaggio per valore, visto che la funzione non modifica **f**
- d) Nessuna delle altre risposte è corretta

2. (3, -.5) Il seguente programma:

```
int &fun(int &a) { a *= 2; return a; }
main()
{
    int p1 = 3;
    int &p2 = fun(p1);
    int &p3 = fun(p1);
    cout<<p1<<' '<<p2<<' '<<p3<<endl;
}
```

- a) stampa 12 12 12
- b) stampa 3 6 12
- c) stampa 6 6 12
- d) nessuna delle altre risposte è vera

3. (3, -.5) Si consideri l'invocazione della seguente funzione, e si supponga che l'array **a** abbia dimensione uguale al valore dell'argomento **size**, e che $v > 0$ e $size > 0$:

```
int fun(const int a[], int v, int size)
{
    int count = 0, i = 0 ;
    while (i < size)
        if (a[i++]%v == 0) continue;
        else count++;
    return count ;
}
```

- a) se l'array **a** contiene almeno un elemento multiplo di **v** la funzione ritorna l'indice del primo di tali elementi, altrimenti ritorna il valore **size**;
- b) se l'array non contiene nessun multiplo di **v**, allora il ciclo **while** è infinito;
- c) nessuna delle altre risposte è vera;
- d) la funzione ritorna il numero di elementi non multipli di **v** all'interno dell'array **a**.

**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 il seguente programma:

```
void scrivi(ofstream &f) { f<<"prova" ; f.close() ; }
main()
{
    ofstream of("dati.txt") ; scrivi(of) ;
    ifstream is("dati.txt") ; char buf[256] ; is>>buf ;
    cout<<buf ;
}
```

- a) il programma stampa **prova** su *stdout*
- b) quando si riapre il file *dati.txt* in lettura all'interno della funzione *main* non è garantito che il suo contenuto sia stato aggiornato
- c) quando si riapre il file *dati.txt* in lettura, quest'ultimo contiene una sequenza di byte, ciascuno dei quali contiene a sua volta il codice di uno dei caratteri precedentemente scritti nel file stesso mediante la funzione *scrivi*
- d) l'invocazione della funzione *f.close()* nella funzione *scrivi* è necessaria per provocare l'aggiornamento del file *dati.txt* prima della sua riapertura in lettura nel programma

5. Assumendo che **rand_num()** sia una funzione che genera un numero intero casuale, il seguente programma:

```
void fun(unsigned int n)
{
    int *a = new int[n]; double media = 0;
    for (int i = 0 ; i < n; i++){ a[i] = rand_num() ; media += a[i]; }
    media /= n;
    for (int i = 0 ; i < n ; i++)
        if (a[i] > media) cout<<a[i]<<endl;
}
main(){ for (;;) fun(10); }
```

- a) ad ogni invocazione della funzione **fun()**, stampa i valori superiori alla media tra i dieci che ha generato a caso in tale funzione
- b) se non terminato dall'utente, va avanti all'infinito
- c) consuma progressivamente tutta la memoria principale del sistema, a meno che non venga terminato prima
- d) non effettua mai accessi in memoria che ne causano la terminazione

6. Nel seguente programma:

```
1:int a = 1;
2:
3:fun()
4:{
5:    int i ;
6:    for (i = 0 ; i < 5 ; i++) {
7:        double a = i / 2.0 ;
```

```

8:      cout<<a * i ;
9:      }
10:     cout<<a + i ;
11:}
12:
13:main()
14:{
15:   double b = 1.4 ;
16:   fun() ;
17:   cout<<a + b<<endl ;
18:}

```

- a) la variabile **b** definita alla riga 15 ha tempo di vita pari all'intero programma;
- b) la variabile **a** definita alla riga 1 non ha visibilità pari all'intero programma;
- c) la variabile **a** definita alla riga 1 non è più visibile alla riga 17 dopo l'esecuzione della funzione **fun**;
- d) la variabile **a** definita alla riga 1 è visibile alla riga 10.

7. Dato il seguente programma e supponendo che il valore massimo memorizzabile in un **char** sia 127, e che l'oggetto **cout** sia configurato per stampare solo la parte intera, senza la virgola, nel caso di numeri double senza parte decimale:

```

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**

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. (6 pt) Spiegare in che modo l'implementazione del tipo di dato *double* permette di rappresentare numeri decimali, ed in che modo permette di rappresentare numeri molto più grandi degli altri tipi di dato. Spiegare infine perché il tipo di dato *double* non può memorizzare qualsiasi numero reale, e fornire un esempio a supporto di tale spiegazione.

9. (6 pt) Senza utilizzare funzioni di libreria, scrivere una funzione che prende in ingresso due stringhe e ritorna vero se la seconda stringa è uguale alla prima o è un anagramma della prima (ossia la seconda stringa, pur non essendo uguale alla prima, contiene tutti e soli i caratteri della prima). Ad esempio la funzione ritorna vero per le stringhe “*diana*” e “*nadia*”. Si ottiene il punteggio massimo se si realizza questa funzione a costo lineare rispetto alla lunghezza delle stringhe.

Risposte Prova scritta di Programmazione I
17 febbraio 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					3	-0,5
4						
5						
6						
7						

Risposta alla domanda 8 (6 pt):

Risposta alla domanda 9 (6 pt):

Pagina extra - 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					3	-0,5
4						
5						
6						
7						

Risposta alla domanda 8 (6 pt):

Risposta alla domanda 9 (6 pt):