

Programmazione I

Prova scritta - 10 luglio 2012 - 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:

```
int c ;
int precedente() { return c-- ; }
```

```
main() {
    cin>>c ;
    int d = precedente() ;
    int e = precedente() ;
    if (e < d) cout<<"Minore" ;
}
```

- a) per qualsiasi numero intero immesso dall'utente, il programma stampa sempre **Minore**
- b) il programma contiene un errore di gestione della memoria
- c) nessuna delle altre risposte è vera
- d) per qualsiasi numero intero negativo immesso dall'utente, il programma non stampa nulla

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

```
void fun(int &) ;
int global ;
main()
```

```
{
    int &ref = global;
    ref = 5 ;
    fun(ref) ;
}
```

```
void fun(int &local)
{
    local++ ;
    cout<<local<<" "<<global ;
}
```

- a) Se eseguito, il programma stampa **6 5**
- b) Se eseguito, il programma stampa **6 6**
- c) Nessuna delle altre risposte è vera
- d) Il programma causa un errore a tempo di compilazione

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

```
main()
{
    int a ; cin>>a ; bool b = a > 1 ;
    if ( (a > 0) == (! b) )
        cout<<a ;
}
```

- a) Se l'utente immette 1 su *stdin* il programma stampa 1
- b) Il programma contiene un errore di gestione della memoria
- c) Nessuna delle altre risposte è vera
- d) Qualsiasi valore l'utente immetta su *stdin*, il programma non stampa nulla

4. (3, -0.5) Il seguente programma

```
main()
{
    bool finito = false ; int i = 0 ;
    while( ! finito ) {
        cout<<i<<" " ;
        if (i == 2)
            finito = true ;
        cout<<i<<" " ;
        i++ ;
    }
}
```

- a) stampa 0 0 1 1 2 2
- b) stampa 0 0 1 1 2
- c) stampa 0 0 1 1
- 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: -0.5
- Una risposta lasciata in bianco viene calcolata: 0

5. Dato un *array* così dichiarato

```
int a[5] ;
```

all'interno di una funzione,

- a) l'*array* continua ad occupare spazio finché non si dealloca la memoria da esso occupata mediante l'operatore **delete**;
- b) l'espressione **a** può essere utilizzata per passare l'*array* ad una funzione;
- c) l'istruzione **a[-1] = 0**; è corretta ed assegna il valore 0 all'elemento che precede il primo elemento dell'*array*;
- d) l'istruzione **a[0]=-1**; è corretta ed assegna il valore -1 al primo elemento dell'*array*.

6. Indicare quale/i delle seguenti affermazioni sulle stringhe è/sono corrette;

- a) il terminatore di stringa non è indispensabile nel caso di un'allocazione dinamica della stringa
- b) il carattere '\n' non può essere utilizzato per indicare il fine stringa
- c) le stringhe statiche non vengono memorizzate nello *heap*
- d) lo spazio occupato in memoria dall'*array* di caratteri utilizzato per rappresentare una stringa dipende dal numero corrente di caratteri della stringa

7. Dato il seguente programma

```
struct parola {char *s ; int n ;} ;

main()
{
    parola v[3] ;
    for (int i = 0 ; i < 3 ; i++) {
        v[i].s = new char[100] ;
        cin>>v[i].s ;
        v[i].n = strlen(v[i].s) ;
        cout<<v[i].s<<" "<<v[i].n<<endl ;
    }
}
```

e definita una parola come una sequenza di caratteri non separati da spazi:

- Purché l'utente non inserisca parole più lunghe di 100 caratteri, il programma legge correttamente da *stdin* tre parole e stampa su *stdout* ciascuna delle tre parole seguita dalla rispettiva lunghezza
- Purché l'utente non inserisca parole più lunghe di 99 caratteri, il programma legge correttamente da *stdin* tre parole e stampa su *stdout* ciascuna delle tre parole seguita dalla rispettiva lunghezza
- Nessuna delle altre risposte è vera
- Anche se il programma non dealloca esplicitamente gli oggetti che alloca dinamicamente durante la sua esecuzione, la memoria occupata da tali oggetti è restituita ugualmente al sistema operativo al termine del programma

8. Data l'istruzione `cin>>i` ; ove `i` è una variabile di tipo `int` correttamente definita:

- Se, all'esecuzione di tale istruzione, lo *stdin* già contiene una sequenza di caratteri che rappresenta un numero intero, il programma non si blocca in attesa dell'utente, ma prova immediatamente a memorizzare tale numero nella variabile `i`
- All'esecuzione di tale istruzione il programma si blocca sempre in attesa che l'utente invii dei caratteri sullo *stdin*
- Se, all'esecuzione di tale istruzione, lo *stdin* contiene una sequenza di caratteri che rappresenta un numero intero, seguita da uno spazio ed infine da una sequenza di caratteri che non rappresenta alcun numero, il `cin` non va in stato di errore
- Nel caso in cui tale istruzione sia eseguita con successo, ossia venga memorizzato correttamente un numero intero letto da *stdin* nella variabile `i`, lo *stdin* è svuotato di tutto il suo contenuto

9. Dato il seguente programma

```
int g = 1 ;
double fun(int n)
{ g = g + 1 ; return n * 3 ; }

main()
{
    const double g = 3 ;
    double l = static_cast<int>(fun(g / 2)) / 2 ;
    cout<<static_cast<int>(l * g) ;
}
```

- Se eseguito, il programma stampa 3
- La conversione esplicita presente nell'espressione a destra dell'inizializzazione nella penultima istruzione della funzione *main* non causa alcuna perdita di informazione
- Se non fosse stata presente la conversione esplicita nell'espressione a destra dell'inizializzazione nella penultima istruzione della funzione *main* il programma avrebbe stampato un valore diverso

- d) Il programma causa un errore a tempo di compilazione, perché la prima istruzione della funzione *fun* prova a modificare la costante *g* definita nella prima istruzione 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**

10. (4 pt.) Cosa scrive su *stdout* il seguente programma?

```
void fun(char [], int);

main()
{
    char string[] = "mare";
    cout<<string<<endl;
    fun(string, strlen(string));
    cout<<string<<endl;
}

void fun(char s[], int n)
{
    int i = 0, j = n - 1 ;
    while (s[i] != '\0'){
        s[i] = s[j] ;
        i++ ; j-- ;
    }
}
```

11. (5 pt.) Scrivere una funzione che prenda in ingresso un *array* di interi **a** e la sua dimensione, e ritorni un *array* contenente i seguenti valori interi: il numero di elementi dell'*array* **a**, gli elementi dell'*array* **a** in ordine inverso, la somma dei valori degli elementi dell'*array* **a** (trascurare problemi di *overflow*). Esempio: dato in ingresso l'*array* {1, 5, 3}, la funzione restituisce l'*array* {3, 3, 5, 1, 9}.

Programmazione I

Prova scritta - 10 luglio 2012

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 un 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					3	-0,5
5						
6						
7						
8						
9						

Risposta alla domanda 10 (4 pt):

Risposta alla domanda 11 (5 pt):

Programmazione I

Prova scritta - 10 luglio 2012

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 un 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					3	-0,5
5						
6						
7						
8						
9						

Risposta alla domanda 10 (4 pt):

Risposta alla domanda 11 (5 pt):