Programmazione I
Prova scritta - 26 febbraio 2014 - 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:
```
int fun(const int a[]) { return a[0] + 1 ; }
main() { int b[2] = {2, 3} ; cout<<fun(b) ; }
```
   a) Quando la funzione **fun** è invocata, il valore di ciascuno degli elementi dell'*array* **b** (parametro attuale) è copiato nel corrispondente elemento dell'*array* **a** (parametro formale)
   b) Quando la funzione **fun** è invocata, l'indirizzo del (primo elemento) dell'*array* **b** (parametro attuale) è copiato nel parametro formale **a**
   c) Se la funzione **fun** contenesse istruzioni che modificano il valore di un elemento dell'*array* **a** (parametro formale) sarebbe segnalato un errore a tempo di esecuzione, in particolare l'errore sarebbe segnalato nel momento in cui si tenta di eseguire tale istruzione
   d) Nessuna delle altre risposte è vera

2. **(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(int a[], int v, int size)
{
   int count = 0, i = 0 ;

   while (i < size)
        if (a[i++]%v == 0)
             break;
        else
             count++ ;
   return count ;
}
```

   a) se l'*array* **a** contiene almeno un elemento non multiplo di **v** la funzione ritorna l'indice del primo di tali elementi, altrimenti ritorna il valore **size**;
   b) la funzione ritorna il numero di elementi consecutivi non multipli di **v** a partire dall'inizio dell'array **a**;
   c) se l'array non contiene nessun multiplo di **v**, il ciclo **while** è infinito;

d) nessuna delle altre risposte è corretta

3. **(1, -0.5)** In C/C++, una stringa implementata mediante *array* di caratteri:
   a) va sempre allocata dinamicamente;
   b) può essere copiata in un'altra stringa con una istruzione di assegnamento
   c) nessuna delle altre risposte è vera;
   d) deve sempre essere terminata dal carattere di fine stringa '**\0'.**

4. **(3, -0.5)** Dato il seguente programma

```
main()
{
 int i = 10 ; char s[10] ; ifstream f("dati.txt");
 f>>i ; f>>s ;
 cout<<i<<" "<<s<<endl ;
}
```

   e supponendo che il file *dati.txt* esista e contenga la sequenza di caratteri **2 prova** (c'è uno spazio tra il carattere 2 e la stringa *prova*) seguiti dal carattere *newline*:
   a) Il programma stampa
      2 prova
   b) Il programma memorizza il valore 2 nella variabile *i*, poi si blocca indefinitamente sull'istruzione **f>>s ;** perché manca il carattere *newline* tra il carattere 2 e la stringa *prova* all'interno del file *dati.txt*
   c) Nessuna delle altre risposte è corretta
   d) L'*ifstream f* va in stato di errore sull'istruzione **f>>s ;**

PARTE 2 – (POSSIBILI) RISPOSTE MULTIPLE - Ogni domanda può avere <u>da una a quattro</u> risposte CORRETTE.

- **Ogni risposta esatta viene calcolata: +1**
- **Ogni risposta errata viene calcolata: -0.5**
- **Una risposta lasciata in bianco viene calcolata: 0**

5. Data la seguente definizione:

```
char c = static_cast<int>('d') ;
```

   a) all'atto dell'inizializzazione si ha perdita di informazione
   b) si inizializza **c** con il valore della costante carattere **'d'**
   c) il compilatore deve effettuare una conversione implicita per realizzare l'inizializzazione
   d) se si tentasse di stampare il valore di **c** mediante **cout<<c;** si stamperebbe il codice del carattere **d**

6. 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) Nessuna delle altre risposte è vera

7. Definita una parola come una sequenza di caratteri non separati da spazi e ricordando che l'operatore >> di default legge una parola alla volta da *stdin* se il suo secondo operando è di tipo *array* di caratteri, il seguente programma:

```
#include <cstring>
struct parola {char *stringa ; int lun ;} ;
main()
{
     parola v ; cin>>v.stringa ; v.lun = strlen(v.stringa) ;
     cout<<v.stringa<<" "<<v.lun<<endl ;
}
```

   a) legge correttamente da *stdin* una parola e stampa su *stdout* tale parola seguita dalla rispettiva lunghezza
   b) contiene un errore di gestione della memoria
   c) nessuna delle altre risposte è vera
   d) potrebbe essere terminato forzatamente prima di aver stampato la parola

8. Dato un ciclo **for** o **while**,
   a) mediante l'istruzione **break** si esce dal ciclo più interno contenente l'istruzione e da altri eventuali cicli esterni;
   b) mediante l'istruzione **break** si esce solo dal ciclo più interno contenente l'istruzione;
   c) mediante l'istruzione **continue** si termina l'esecuzione del ciclo più interno contenente l'istruzione e si continua l'esecuzione degli eventuali cicli più esterni;
   d) mediante l'istruzione **continue** non si termina l'esecuzione del ciclo che contiene l'istruzione.

9. Quali delle seguenti affermazioni sono vere?
   a) La funzione **printf** della libreria standard del C ha un numero di argomenti variabile
   b) Nella funzione **scanf** della libreria standard del C, sbagliare il tipo o l'indirizzo della variabile in cui si desidera scrivere il valore letto da *stdin* può causare corruzione della memoria
   c) La funzione **printf** della libreria standard del C determina automaticamente il formato con cui stampare ciascuno oggetto passato come argomento (dopo la stringa *format*) in base al tipo dell'oggetto stesso
   d) In C non esiste il passaggio di un oggetto per riferimento, ma si può emularlo passando l'indirizzo dell'oggetto stesso

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. **(5 pt)** Descrivere sintassi e semantica dell'operatore logico || in non più di sei righe, più un'eventuale tabella.

11. **(6 pt)** Senza utilizzare funzioni o oggetti di libreria per l'input formattato da stringhe, scrivere una funzione che prenda in ingresso una stringa da interpretare come la rappresentazione in base 10 di un numero naturale, e, se tutti i caratteri della stringa sono cifre in base 10, ritorni tale numero. Se, ad esempio, viene passata la stringa “34”, la funzione ritorna il numero 34. La funzione ritorna invece un'indicazione di errore nel caso in cui almeno uno dei caratteri della stringa non sia una cifra in base 10. Trascurare problemi di *overflow*. Si ottiene il punteggio massimo se: 1) non si utilizza il numero di caratteri della stringa nel codice, e 2) si scorre la stringa una sola volta, e per valori crescenti degli indici.

Programmazione I
Prova scritta - 26 febbraio 2014

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					**1**	**-0,5**
4					**3**	**-0,5**
5						
6						
7						
8						
9						

Risposta alla domanda 10 (5 pt):

Risposta alla domanda 11 (6 pt):

Programmazione I
Prova scritta - 26 febbraio 2014

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/**	
	A	**B**	**C**	**D**	**Penalità**	
1					**3**	**-0,5**
2					**3**	**-0,5**
3					**1**	**-0,5**
4					**3**	**-0,5**
5						
6						
7						
8						
9						

Risposta alla domanda 10 (5 pt):

Risposta alla domanda 11 (6 pt):