

Test di programmazione 1 - [3 Febbraio 2023] [Canale A-E]

Prova in itinere di Programmazione 1 (9 CFU), 03-02-2023

Corso di Studio in Informatica 2022-2023

Dipartimento di Matematica e Informatica

Università di Catania

Istruzioni:

- Il test presenta 20 domande a risposta multipla.
- È necessario rispondere a tutte le domande per completare il test.
- Solo una delle risposte è corretta per ciascuna domanda.
- La durata della prova è di 30 minuti.
- La prova si considera superata se il candidato risponde correttamente a 15 domande sulle 20 proposte.

* Obbligatoria

* Questo modulo registrerà il tuo nome, inserire il nome.

Dati personali

1. Nome *

2. Cognome *

3. Matricola *

Test

4. Il linker: * (1 punto)

- ☒ Permette di ottenere una singola unita' eseguibile a partire da uno o piu' moduli oggetto.
- ☐ Traduce un programma C in codice macchina.
- ☐ esegue il programma una istruzione alla volta
- ☐ si occupa delle direttive del tipo "#define [...]"

5. Siano a e b due variabili di tipo int. Il risultato della seguente espressione: $\text{rand()} \% (b-a+1) + a$ * (1 punto)

- ☒ sara' un numero intero pseudo-casuale in [a,b]
- ☐ sara' un numero intero pseudo-casuale in [a, b+1]
- ☐ sara' un numero intero pseudo-casuale in [a, b-a]
- ☐ sara' un numero double pseudo-casuale in [a+1, b]

6. L'operatore di dereferenziazione e': * (1 punto)

- ☐ &
- ☐ >>
- ☒ *
- ☐ +

7. La seguente codifica di controllo (sia N un intero positivo): for(int i=1; i<N+1; i++) * (1 punto)

- ☐ produce N-1 iterazioni
- ☐ rappresenta un ciclo che non termina
- ☒ produce N iterazioni
- ☐ produce N+1 iterazioni

$N = 2$ \downarrow
 $2 + 1 = 3$
 $i = 1$
 $i = 2$ $> N - 1 \text{ iter.}$

8. La seguente direttiva: #define MAX(a,b) (a>b ? a : b) * (1 punto)

- ☐ non migliora l'efficienza del codice rispetto alla definizione di una funzione
- ☐ permette di definire una funzione senza specificare il tipo di ritorno
- ☐ viene compilata dopo ogni altra istruzione
- ☒ il preprocessore sostituisce tutte le occorrenze di MAX(..., ...) con l'espressione che costituisce l'ultima parte della direttiva

9. E' corretto memorizzare il numero 1234.1234³⁶⁷⁸ in una variabile di tipo * (1 punto)

- ☒ double
- ☐ int
- ☐ float
- ☐ long int

\rightarrow \approx cifra

10. Sia x una variabile double. L'espressione $(\text{int})(x + 0.5)$ * (1 punto)

- ☐ rappresenta la parte intera di $x + 1$
- ☐ rappresenta il valore assoluto di x
- ☒ rappresenta l'arrotondamento di x all'intero più vicino
- ☐ rappresenta la parte intera di x

11. Un esempio di operatore unario è * (1 punto)

- ☐ un operatore relazionale
- ☐ un operatore di assegnazione
- ☒ un operatore di decremento
- ☐ un operatore logico

12. La parola chiave struct: * (1 punto)

- ☐ permette di estendere un tipo primitivo con ulteriori informazioni
- ☐ permette di definire alias di tipi già definiti
- ☐ permette di definire un campo di bit
- ☒ permette di aggregare un insieme di variabili sotto un unico nome

13. L'algoritmo denominato Bubble sort * (1 punto)

- ☐ necessita di operare su una copia dei dati
- ☒ opera mediante una successione di scambi (swap) su coppie di dati
- ☐ non permette di ordinare stringhe in ordine lessicografico
- ☐ e' un algoritmo di ordinamento per dati che siano numeri interi

14. Una funzione con il seguente prototipo: void f(const int *ptr); * (1 punto)

- ☒ non potra' modificare il dato puntato da ptr
- ☐ opera su una copia, sullo stack, dell'intero dato puntato da ptr
- ☐ potra' modificare il dato puntato dal parametro formale ptr
- ☐ nessuno di questi

15. La funzione di libreria fopen(): * (1 punto)

- ☐ permette di riposizione l'indicatore di posizione del file all'inizio del file stesso
- ☐ permette di aprire un file con modalita' sola scrittura oppure sola lettura
- ☒ permette di aprire un file e di associare uno stream per operare sul file stesso
- ☐ il tipo di ritorno e' void

16. Sia definita una variabile s nel seguente modo:

```
char *s = "Pippo";
```

indicare quale operazione non sara' ammessa dal compilatore: * (1 punto)

- ☐ printf("%c", s[3]);
- ☒ s[2] = 'x';
- ☐ printf("%s", s);
- ☐ for(int i=0; i<strlen(s); i++) printf("%c -->", s[i]);

17. La tecnica di ricerca binaria (o dicotomica) * (1 punto)

- ☐ può essere utilizzato solo su un array
- ☒ produrra' un risultato corretto solo se l'array e' ordinato
- ☐ si puo' applicare solo su array ordinati con algoritmo bubble sort
- ☐ e' piu' efficiente su array non ordinati

18. Sia definita la seguente funzione:

```
double *f(int a, int b){  
    double x = a+(double) b/2;  
    return &x;  
}
```

→ ALLOCATA LOCALMENTE.
** (1 punto) → NON POSSO PASSARLA*

- ☐ La funzione contiene codice che potrebbe provocare un grave errore a tempo di esecuzione; viene segnalato dal compilatore con un errore;
- ☐ L'espressione calcolata dalla funzione non consente di ottenere un risultato preciso;
- ☒ La funzione contiene codice che potrebbe provocare un grave errore a tempo di esecuzione; viene segnalato dal compilatore con un warning;
- ☐ I parametri formali della funzione dovrebbero essere di tipo double;

19. Sia data la seguente dichiarazione:

```
struct record *ptr = malloc(sizeof(struct record));
```

Indicare la forma corretta per l'accesso al campo data (sia un int) del record. *

(1 punto)

- ☐ ptr >> data = 23;
- ☐ (*ptr)->data = 23;
- ☒ ptr->data = 23;
- ☐ ptr.data = 23;

20. Nella struttura dati detta coda: * (1 punto)

- ☐ Se la coda e' non ordinata, non e' possibile inserire un dato in maniera corretta.
- ☐ Inserimento e cancellazione avvengono con modalita' LIFO.
- ☒ Inserimento e cancellazione avvengono con modalita' FIFO.
- ☐ L'inserimento di un dato avviene in testa oppure in coda, in base all'ordinamento dei dati.

21. La seguente istruzione:

`int **a = malloc(sizeof(int *)*n);` * (1 punto)

- ☐ permette di ottenere un doppio puntatore ad un array di int
- ☒ permette di allocare un vettore di n puntatori al tipo int
- ☐ permette di allocare una matrice di interi di dimensione n x n
- ☐ permette di allocare un vettore di interi di dimensione n

22. Una ipotetica funzione di inserimento per una struttura dati dinamica che sia una lista concatenata: * (1 punto)

- ☒ potrebbe modificare la testa della lista
- ☐ non potra' ricevere in input una lista vuota;
- ☐ se la lista e' piena, restituira' NULL
- ☐ potra' inserire solo in coda

23. Per definire un record che contiene un intero ed una stringa di al piu' 20 caratteri: * (1 punto)

- ☒ struct record {int dato; char s[21];}
- ☐ struct record {int dato; char s[21] = {0}};
- ☐ struct record {int dato; char s[20]};
- ☐ struct record {int dato; char *s = malloc(sizeof(char)*21)};

Questo contenuto non è stato creato né approvato da Microsoft. I dati che invii verranno recapitati al proprietario del modulo.