Iniziato	martedì, 6 febbraio 2024, 18:00
Stato	Completato
Terminato	martedì, 6 febbraio 2024, 18:17
Tempo impiegato	16 min. 51 secondi
Valutazione	**1,88** su un massimo di 3,00 (**62,5**%)

Domanda 1

Risposta corretta

Punteggio ottenuto 0,25 su 0,25

Il tipo float può essere utilizzato per codificare numeri reali in maniera esatta

Scegli una risposta:

- Vero
- Falso ✔

Il tipo float può essere utilizzato per memorizzare **approssimazioni** di numeri reali

La risposta corretta è 'Falso'.

Domanda 2

Risposta non data

Punteggio max.: 0,25

Considerare le seguenti assegnazioni. Per ciascuna assegnazione, è indicato tra parentesi quadre un possibile valore contenuto nella variabile di destinazione dopo che l'assegnazione è avvenuta.

Selezionare tutte le opzioni in cui tale valore è corretto.

Le variabili utilizzate sono dichiarate come segue:

```
int a=5;
int e=2;
double x=0.8;
double y=1.2;
bool b=true;
```

Per ogni assegnazione, le variabili hanno i valori indicati in inizializzazione.

- a. ((a>e)&&(a>10)) || (a<=5) [false]
- b. b=(a/e*e==a) [true]
- c. y=a/e; [2.5]
- d. e=a+3.2; [8]
- e. b=((a==e)&&(5!=e) [false]

Risposta errata.

Le risposte corrette sono:
e=a+3.2; [8],

b=((a==e)&&(5!=e) [false]

Domanda 3

Risposta errata

Punteggio ottenuto 0,00 su 0,25

FUNZIONI. Indicare l'unica risposta vera.
Nel passaggio dei parametri di funzioni per riferimento

Scegli un'alternativa:

- a. I parametri attuali possono essere variabili, valori o espressioni
- b. Nessuna delle altre scelte
- c. I parametri formali ricevono una copia del contenuto del corrispondente parametro attuale ✖
- d. L'ambito di visibilità dei parametri formali comprende la funzione e il main
- e. Ogni modifica del contenuto del parametro formale effettuata all'interno della funzione, rimarrà memorizzata alla fine dell'esecuzione della funzione stessa all'interno del corrispondente parametro attuale

La risposta corretta è: Ogni modifica del contenuto del parametro formale effettuata all'interno della funzione, rimarrà memorizzata alla fine dell'esecuzione della funzione stessa all'interno del corrispondente parametro attuale

Domanda 4

Risposta corretta

Punteggio ottenuto 0,25 su 0,25

Completare il frammento di codice **reverse** che copia da un array source ad un array dest gli elementi in ordine inverso (assumiamo che entrambi gli array abbiano lunghezza N)

```
for (int i=0; i<N;++i)
    dest[i]= source[N-1-i] ✔ ;
```

Risposta corretta.

La risposta corretta è:
Completare il frammento di codice **reverse** che copia da un array source ad un array dest gli elementi in ordine inverso (assumiamo che entrambi gli array abbiano lunghezza N)

```
for (int i=0; i<N;++i)
    dest[i]=[source[N-1-i]];
```

Domanda 5

Risposta corretta

Punteggio ottenuto 0,25 su 0,25

Assumiamo che num sia una variabile di tipo unsigned int che contiene un numero naturale.
Qual è lo scopo del seguente frammento di codice

```
int contatore=0;
while (num>0) {
    num/=10;
    contatore++;
}
```

a. verifica se il numero è un multiplo di 10000

b. calcolo del numero di cifre che compongono il numero ✔

c. calcolo del numero di 0 del fattoriale

d. fattorizzazione in fattori primi

Risposta corretta.

La risposta corretta è:
calcolo del numero di cifre che compongono il numero

Domanda 6

Risposta errata

Punteggio ottenuto 0,00 su 0,25

Consideriamo array dinamici relizzati come segue

```
struct my_vector{
    int * store ;
    unsigned int size ;
    unsigned int capacity;
};
```

e la funzione

```
my_vector my_vector_constructor() {
    my_vector v;
    v.store = nullptr;
    v.size=0;
    v.capacity=0;
    return v;
}
```

scopo della funzione è (scegliere alternativa che vi paia maggiormente adeguata)

a. la funzione è errata / priva di senso ✖

b. creare un my_vector predisponendo size elementi in memoria primaria (heap)

c. creare un my_vector con size e capacity elementi nello heap

d. creare un my_vector vuoto, che potrà in un secondo momento ricevere un puntatore ad elementi in memoria primaria (heap) tramite una futura assegnazione al campo store

Risposta errata.

La risposta corretta è:
creare un my_vector vuoto, che potrà in un secondo momento ricevere un puntatore ad elementi in memoria primaria (heap) tramite una futura assegnazione al campo store

Domanda 7

Risposta corretta

Punteggio ottenuto 0,25 su 0,25

VECTOR. Qual è lo scopo della funzione qui di seguito?

```
bool funzione(const vector<int> &v, int val) {
    for (unsigned int i=0;i<v.size();++i)
        if (v.at(i)==val)
            return true;
  return false;
}
```

Scegli un'alternativa:

- a. verifica che tutti gli elementi del vector v siano diversi a val
- b. verifica che v contenga esattamente un elemento uguale a val
- c. verifica che tutti gli elementi del vector v siano uguali a val
- d. verifica che v contenga almeno un elemento uguale a val ✔

Risposta corretta.

La risposta corretta è: verifica che v contenga almeno un elemento uguale a val

Domanda 8

Risposta corretta

Punteggio ottenuto 0,25 su 0,25

considerato il tipo di dato

```
typedef struct cell {
   int head;
  cell *next;
} *lista;
```

Qual e' lo scopo della seguente funzione

```
void A(lista &l) {
l=nullptr;
}
```

- a. verifica se una lista è vuota
- b. crea una cella nuova per una lista vuota
- c. crea una lista vuota ✔
- d. cancella tutti gli elementi da una lista

Risposta corretta.

La risposta corretta è:
crea una lista vuota

Domanda 9

Risposta corretta

Punteggio ottenuto 0,25 su 0,25

Completare la funzione di ricerca binaria

```
bool ric_binaria(const int array[N], int elem, int first, int last ) {
    if (first > last) ✔ return false;
    int mid=(first+last)/2;
    if (elem == array[mid]) ✔ return true;
    else{
        if (elem < array[mid])
            return ric_binaria(array, elem, first, mid-1); ✔
        else return ric_binaria(array, elem, mid+1,last); ✔
 }
}
```

Risposta corretta.

La risposta corretta è:
Completare la funzione di ricerca binaria

```
bool ric_binaria(const int array[N], int elem, int first, int last ) {
    [if (first > last)] return false;
    int mid=(first+last)/2;
   [if (elem == array[mid])] return true;
    else{
        if (elem < array[mid])
            return  [ric_binaria(array, elem, first, mid-1);]
        else    return [ric_binaria(array, elem, mid+1,last);]
 }
}
```

Domanda 10

Risposta corretta

Punteggio ottenuto 0,25 su 0,25

Completare la funzione ricorsiva che calcola le potenze n^m

```
int recoursive_power (int n, unsigned int m) {
   if (m==0) return 1;
   if(m==1) return n;
   return [n*recoursive_power(n,m-1);] ✔
}
```

Risposta corretta.

La risposta corretta è:

Completare la funzione ricorsiva che calcola le potenze n^m

```
int recoursive_power (int n, unsigned int m) {
   if (m==0) return 1;
   if(m==1) return n;
   return [n*recoursive_power(n,m-1);]
}
```

Domanda 11

Parzialmente corretta

Punteggio ottenuto 0,13 su 0,25

Consideriamo un'implementazione di array dinamici come quella vista in classe, basata sulla seguente struct

```
struct dynamic_array {
   int * store ;
   unsigned int size ;
};
```

Relativamente alla seguente funzione, quali affermazioni sono vere?

```
void una_funzione(dynamic_array &d, int index, int value) {
   if ((index >= d.size) || (index < 0)) throw SOME_ERROR;
   *(d.store+index)=value;
}
```

- a. la funzione permette di inserire il valore value nella posizione index dell'array
- b. la funzione non e' in grado di gestire il caso in cui index sia out of bound
- c. la funzione solleva un'eccezione se index e' out of bound ✔
- d. la funzione permette di stampare il valore value nella posizione index dell'array
- e. la funzione permette di verificare se il valore value si trova nella posizione index dell'array
- f. la funzione restituisce valore speciale se index e' out of bound

Risposta parzialmente esatta.

Hai selezionato correttamente 1.
Le risposte corrette sono:
la funzione permette di inserire il valore value nella posizione index dell'array,

la funzione solleva un'eccezione se index e' out of bound

Domanda 12

Risposta errata

Punteggio ottenuto 0,00 su 0,25

Qual è il contenuto dell'array v alla fine dell'esecuzione dei seguenti comandi?

```
int v[5]={3,6,9,12,15};
int *p=v;
p=p+1;
*p=4;
```

Scegli un'alternativa:

- a. 3 4 9 12 15
- b. 3 4 4 4 4
- c. 3 6 9 4 15
- d. 3 4 6 9 12 15 ✖

Risposta errata.

La risposta corretta è: 3 4 9 12 15