

Corso di Programmazione I e Laboratorio di Programmazione I

Parte I : Prof. Giulio Giunta
Parte II : Prof. Angelo Ciaramella

A.A. 2020/2021

Studente

- **Cognome:**
- **Nome:**
- **Matricola:** 124/

Quesiti

I quesiti sono finalizzati alla simulazione, da parte dello studente iscritto al corso, di una verifica intercorso e/o di profitto finale con le relative risposte descritte e commentate, sono indicati altresì i riferimenti da consultare del manuale di riferimento.

Quesito n. 1

Scrivere esattamente come mostrato ed eseguire il seguente codice. Verificare e correggere gli errori di compilazione.

```
0
1
2     $include 'stdio'
3
4     void()
5     "Il mio programma in C funziona ?"
6
7     return 0
8
```

Riferimento

- Manuale di Programmazione in C
- Capitolo 1

Soluzione

```
0  #include <stdio.h>
2
4  int main() {
6  printf("Il mio programma in C funziona ? ");
   return 0;
}
```

Quesito n. 2

Scrivere un programma che calcola e stampa il risultato della seguente formula

$$\frac{x^2 + a \cdot x}{b + x} + \frac{c \cdot x + d}{x^3 + x} \cdot e \quad (1)$$

Le costanti sono così definite: $a = 3$, $b = 0.5$, $c = 0.1$, $d = 1$, $e = 5$. x è inserito dall'utente ed è il dato di input del programma.

Riferimento

- Manuale di Programmazione in C
- Capitolo 1

Soluzione

```
0  #include <stdio.h>
   #include <math.h>
2
4  int main() {
6  const int a = 3;
   const float b = 0.5;
   const float c = 0.1;
   const int d = 1;
   const int e = 5;
8
10 double x, espressione;
12 printf("Dammi il valore di x: ");
   scanf("%lf", &x);
14
   espressione = (pow(x,2) + a * x) / (b + x) + (c * x + d) / (pow(x,3) + x) *
   e;
16
   printf("Il risultato dell'espressione con x = %lf e': %lf", x,
   espressione);
18
   return 0;
20 }
```

Quesito n. 3

Scrivere un programma che data una variabile intera x , con valore inserito dall'utente, permette di moltiplicare per 3 il suo contenuto mediante accesso indiretto (usare un puntatore px). Confrontare il risultato ottenuto con accesso alla variabile x in modo diretto e indiretto.

Riferimento

- Manuale di Programmazione in C
- Capitolo 1

Soluzione

```
0  #include <stdio.h>
   #include <math.h>
2
   int main() {
4
   int x;
   int *px;
6
   printf("Dammi il valore di x: ");
   scanf("%d", &x);
8
   px = &x;
10
   *px *= 3;
12
   printf("Il valore di x e' (accesso diretto): %d", x);
14
   printf('\'Il valore di x e\' (accesso indiretto): %d\'', *px);
16
   return 0;
18
   }
20
```

Quesito n. 4

Dato il seguente codice sostituire il ciclo **for** con un ciclo **do-while**.

```
0  #include <stdio.h>
2  int main() {
4
   int i, num;
   double x, s = 0.0;
6
   const int chiave = 10;
8
   printf("Quanti numeri vuoi inserire ?");
   scanf("%d", &num);
10
   for(i = 0 ; i < num ; i++) {
12       printf("\n Inserisci %d-esimo numero: ", i+1);
14       scanf("%lf",&x);
```

```

16     s +=x;
17     }
18     if (s != chiave) printf(" \n La somma e' diversa dalla chiave impostata")
19     ;
20     return 0;
21 }

```

Riferimento

- Manuale di Programmazione in C
- Capitolo 2

Soluzione

```

0     #include <stdio.h>
1     #include <math.h>
2
3     int main() {
4
5         int i=0, num;
6         double x, s = 0.0;
7
8         const int chiave = 10;
9
10        printf("Quanti numeri vuoi inserire ?");
11        scanf("%d", &num);
12
13
14        do{
15            printf("\n Inserisci %d-esimo numero: ", i+1);
16            scanf("%lf",&x);
17
18            s +=x;
19
20            i++;
21
22        }while(i < num);
23
24        if (s != chiave) printf(" \n La somma e' diversa dalla chiave impostata")
25        ;
26
27        return 0;
28    }

```

Quesito n. 4

Dato il seguente codice sostituire il ciclo **while** con un ciclo **do-while**.

```

0  #include <stdio.h>
   #include <stdlib.h>
2
   int main() {
4
       char c;
6       int lower = 0, total = 0;
8
       while(( c = getchar()) != EOF){
           if (c >= 'a' && c <= 'z'){
10                ++lower;
12                total++;
14                fflush(stdin);
           }
16
           printf("Su un totale di %d lettere %d sono minuscole \n", total, lower);
18
       return 0;
   }

```

Riferimento

- Manuale di Programmazione in C
- Capitolo 2

Soluzione

```

0  #include <stdio.h>
   #include <stdlib.h>
2
   int main() {
4
       char c;
6       int lower = 0, total = 0;
8
       do{
           fflush(stdin);
10
           c = getchar();
12
           if (c >= 'a' && c <= 'z'){
14                ++lower;
16                total++;
           }while(c != EOF);
18
           printf("Su un totale di %d lettere %d sono minuscole \n", total - 1,
               lower);
20
       return 0;
   }
22

```

Quesito n. 5

Riscrivere il seguente programma sostituendo la function in C con una procedura equivalente.

```
0 #include <stdio.h>
1 #include <stdlib.h>
2
3 int main() {
4
5     int i;
6     int a, s = 0;
7
8     for(i=0;i<3;i++)
9     {
10         printf("Dammi il %d numero \n", i+1);
11         scanf("%d", &a);
12         s += f(a);
13
14     }
15
16     printf("Il risultato e : %d", s);
17
18     return 0;
19 }
20
21 int f(int a)
22 {
23     static int cnt = 1;
24
25     ++cnt;
26
27     if (cnt % 2 == 0)
28         return a+cnt;
29     else
30         return 0;
31 }
32 }
```

Riferimento

- Manuale di Programmazione in C
- Capitolo 3

Soluzione

```
0 #include <stdio.h>
1 #include <stdlib.h>
2
3 void fp(int *);
4
5 int main() {
6
7     int i;
8     int a, s = 0;
9
10    for(i=0; i<3; i++)
11    {
12        printf("Dammi il %d numero \n", i+1);
```

```

14     scanf("%d", &a);
16     fp(&a);
18     s += a;
20     printf("Il risultato e : %d", s);
22 return 0;
24 }
26 void fp(int *a)
28 {
30     static int cnt = 1;
32     ++cnt;
34     if (cnt % 2 == 0)
36         *a += cnt;
38     else
39         *a = 0;
40 }

```

Quesito n. 6

Riscrivere il seguente programma sostituendo la procedura con una function equivalente.

```

0 #include <stdio.h>
1 #include <stdlib.h>
2
3 void f_proc(int *, int *, int *);
4
5 int main() {
6
7     int a, b, c;
8
9     printf("Dammi a, b e c \n");
10    scanf("%d%d%d",&a, &b, &c);
11
12    f_proc(&a, &b, &c);
13
14    printf("\n I nuovi valori di a, b e c sono: %d - %d - %d \n", a, b, c);
15
16    return 0;
17 }
18
19 void f_proc(int *a, int *b, int *c) {
20     if (*a <= *b)
21         *c += *a;
22     else
23         *c += *b;
24 }

```

Riferimento

- Manuale di Programmazione in C

- Capitolo 3

Soluzione

```
0 #include <stdio.h>
1 #include <stdlib.h>
2
3 int f_proc_f(int, int, int);
4
5 int main() {
6     int a, b, c;
7
8     printf("Dammi a, b e c \n");
9     scanf("%d%d%d", &a, &b, &c);
10
11     c = f_proc_f(a, b, c);
12
13     printf("i nuovi valori di a, b e c sono: %d - %d - %d \n", a, b, c);
14
15     return 0;
16 }
17
18 int f_proc_f(int a, int b, int c) {
19     if (a <= b)
20         return c + a;
21     else
22         return c + b;
23 }
24 }
```

Quesito n. 7

Scrivere una funzione che prende in input due array di *float* di dimensioni n e restituisce un array composto dai due array dove nelle posizioni dispari sono contenuti i valori del primo array e nelle posizioni pari quelle del secondo.

Riferimento

- Manuale di Programmazione in C
- Capitolo 4

Soluzione

```
0 void f_array(float *a, float *b, int n, float *c){
1
2     int i, j = 0;
3
4     for (i = 1; i < 2 * n; i += 2)
5     {
6         c[i-1] = b[j];
7         c[i] = a[j];
8         j++;
9     }
10 }
```



```
12 }
```

Quesito n. 8

Scrivere una funzione che prende in input una matrice e restituisce in output la matrice trasposta (ovvero la matrice in cui la j -esima colonna è la j -esima riga della matrice data, per $j = 1, 2, \dots, n$)

Riferimento

- Manuale di Programmazione in C
- Capitolo 4

Soluzione

```
0 void Trasposta(float *A, int n_col, int m_rig, float *B){
2     int i, j;
4     for (i = 0; i < m_rig; i++)
5     {
6         for (j = 0; j < n_col; j++)
7         {
8             *(B + m_rig * j + i) = *(A + n_col * i + j);
9         }
10    }
```

Quesito n. 9

Sviluppare una function C che, dati come parametri di input un array 2D di *float*, il numero delle righe, il numero delle colonne, un int p e un int q (con int p minore di int q), determina e restituisce come parametro di output la somma degli elementi che hanno gli indici di riga e di colonna compresi tra int p e int q .

Riferimento

- Manuale di Programmazione in C
- Capitolo 4

Soluzione

```
0 float Somma_p_q(float *A, int m_rig, int n_col, int p, int q){
2   int i, j;
4   float somma = 0.0;
6   if ((p > q) || (p > m_rig) || (q > n_col))
8       return 0;
9   else
10      for (i = 0; i < p; i++)
11          for (j = 0; j < q; j++)
12              somma += *(A + n_col * i + j);
14  return somma;
}
```

Quesito n. 10

Scrivere una function C che determina l'appartenenza di un dato (chiave) ad un array 2D di *float*.

Riferimento

- Manuale di Programmazione in C
- Capitolo 4

Soluzione

```
0 int appartiene_a2DF(float chiave, float a[][100], int n, int m)
2 {
3   int i, j;
4   for(i=0; i<n; i++)
5       for(j=0; j<m; j++)
6           if (chiave == a[i][j])
7               return 1;
8   return 0;
9 }
```

Quesito n. 11

Scrivere una function C che determina il valore massimo e la sua posizione in un array di *float*.

Riferimento

- Manuale di Programmazione in C
- Capitolo 4

Soluzione

```
0 void max_val_ind(float a[], int n, float *max_array, int *i_max)
2 {
4     int i;
6     *max_array = a[0];
6     *i_max = 0;
8     for(i = 1; i < n; i++)
10        if (a[i] > *max_array)
12            {
14                *i_max = i;
14                *max_array = a[i];
16            }
16 }
```

Quesito n. 12

Scrivere una function C per determinare l'uguaglianza tra due insiemi di cardinalità n .

Riferimento

- Manuale di Programmazione in C
- Capitolo 4

Soluzione

```
0 int uguaglianza_insiemiF(float a[], float b[], int n)
2 {
4     int i;
4     i = 0;
6     for(i=0; i<n; i++)
8         if (!appartiene_aF(a[i], b, n))
10            return 0;
12    return 1;
14 }
16 int appartiene_aF(float chiave, float a[], int n)
18 {
12     int i;
14     for(i=0; i<n; i++)
16         if (chiave == a[i])
18            return 1;
18    return 0;
18 }
```

Quesito n. 13

Scrivere una function C che data una matrice di *double* di dimensione $m \times n$ restituisce la media delle righe (usare la notazione a puntatori per l'accesso alla matrice).

Riferimento

- Manuale di Programmazione in C
- Capitolo 4

Soluzione

```
0 void calcola_media_righe_ap2D(double *A, int n_col, int m, int n, double
1   media_riga[])
2 {
3     int i,j;
4     double somma = 0.0;
5
6     for (i = 0; i < m; i++)
7     {
8         somma = 0.0;
9
10        for( j = 0; j < n; j++)
11            somma+= *(A + i*n_col +j);
12
13        media_riga[i] = somma/n;
14    }
15 }
16 }
```

Quesito n. 14

Scrivere una function C che prende in input due array ordinati di *char* e restituisce un terzo array ordinato ottenuto mediante la fusione dei due (**algoritmo di fusione**).

Riferimento

- Manuale di Programmazione in C
- Capitolo 4

Soluzione

```
0 void fusioneC(char a[], int n_a, char b[], int n_b, char c[])
1 {
2     int i_a = 0, i_b = 0, i_c = 0;
3
4     while(i_a < n_a && i_b < n_b)
```

```

6   {
8       if(a[i_a] < b[i_b])
           c[i_c++] = a[i_a++];
10      else
           c[i_c++] = b[i_b++];
12      }
14      while(i_a < n_a)
16      {
           c[i_c++] = a[i_a++];
18      }
20      while(i_b < n_b)
22      {
           c[i_c++] = b[i_b++];
24      }
}

```

Quesito n. 15

Scrivere una function C che dato un testo e una chiave restituisce l'indice iniziale della sottostringa del testo con il *matching migliore* (algoritmo di **best matching**)

Riferimento

- Manuale di Programmazione in C
- Capitolo 5

Soluzione

```

0   int matching_migliore(char *chiave, char *testo)
2   {
4       int i, n, m, punteggio_max = 0, punteggio, indice = -1;
6       n = strlen(chiave);
8       m = strlen(testo);
10      for(i = 0; i < m - n ; i++)
12      {
           punteggio = punteggio_matching(chiave, &testo[i], n);
14          if (punteggio > punteggio_max)
16          {
               punteggio_max = punteggio;
               indice = i;
18          }
20      }
22      return indice;
24  }

```

```

26 int punteggio_matching(char *a, char *b, int n)
27 {
28     int i, n_caratteri_uguali = 0;
29     for(i = 0 ; i < n ; i++)
30         if(a[i] == b[i])
31             n_caratteri_uguali++;
32     return n_caratteri_uguali;
33 }
34

```

Quesito n. 16

Scrivere una function C che dato un testo restituisce il numero di caratteri, di parole e di linee nel testo.

Riferimento

- Manuale di Programmazione in C
- Capitolo 4

Soluzione

```

0 void conta_tutto(char *testo, int *conta_c, int *conta_p, int *conta_l)
1 {
2     int i = 0, flag = 0;
3
4     *conta_c = 0;
5     *conta_p = 0;
6     *conta_l = 0;
7
8     while(testo[i] != '\0')
9     {
10        (*conta_c)++ ;
11
12        if(testo[i] == '\n')
13            (*conta_l)++;
14
15        if(testo[i] == ' ' || testo[i] == '\n' || testo[i] == '\t')
16            flag = 0;
17        else
18            if (flag == 0)
19            {
20                flag = 1;
21                (*conta_p)++;
22            }
23        i++;
24    }
25    (*conta_l)++;
26 }

```

Quesito n. 17

Scrivere una function C che dato un testo restituisce il numero di parole che finiscono in **are**.

Riferimento

- Manuale di Programmazione in C
- Capitolo 5

Soluzione

```
0 int conta_are(char *testo)
2 {
4     char separatori[] = {' ', '\n', '\t', '\0'}, *token = NULL;
6     int cont = 0;
8     token = strtok(testo, separatori);
10    while( token != NULL)
12    {
14        if (strncmp(&token[strlen(token)-3], "are", 3) == 0)
16            cont++;
18        token = strtok(NULL, separatori);
19    }
20    return cont;
21 }
```

Quesito n. 18

Scrivere una function C che dato un array di *char* restituisce l'array ordinato in ordine crescente (usare l'algoritmo di **ordinamento per inserimento**).

Riferimento

- Manuale di Programmazione in C
- Capitolo 5

Soluzione

```
0 void ord_inser(char a[], int n)
2 {
4     int i, j;
6     char el_da_ins;
8     for (i = 1; i < n; i++)
10    {
12        el_da_ins = a[i];
14    }
```

```

10     j = i - 1;
12     while (j >= 0 && el_da_ins < a[j])
14     {
16         a[j+1] = a[j];
18         j--;
20     }
22     a[j + 1] = el_da_ins;
24 }

```

Quesito n. 18

Scrivere una function C che dato un array di *char* restituisce l'array ordinato in ordine crescente (usare l'algoritmo di **selezione di minimo**).

Riferimento

- Manuale di Programmazione in C
- Capitolo 5

Soluzione

```

0 void ord_sel_min(char a[], int n)
2 {
4     int i;
6     for (i = 0 ; i < n - 1; i++)
8     {
10         min_val_ind(&a[i], n - i, &min_array, &indice_min);
12         scambiare_c(&a[i], &a[indice_min + i]);
14     }
16 }
18 void min_val_ind(char a[], int n, char *min_array, int *i_min)
20 {
22     int i;
24     *min_array = a[0];
26     *i_min = 0;
28     for (i = 1 ; i < n ; i++)
30         if (a[i] < *min_array)
32             *min_array = a[i];

```



```

28         *i_min = i;
29     }
30
31 }
32
33
34 void scambiare_c(char *a, char *b)
35 {
36     char temp;
37
38     temp = *a;
39
40     *a = *b;
41
42     *b = temp;
43
44 }

```

Quesito n. 19

Scrivere una function C che dato un array di *char* restituisce l'array ordinato in ordine crescente (usare l'algoritmo di **selezione di massimo**).

Riferimento

- Manuale di Programmazione in C
- Capitolo 5

Soluzione

```

0
1 void ord_sel_max(char a[], int n)
2 {
3     int i;
4
5     for (i = n - 1 ; i > 0; i--)
6         scambiare_c(&a[i], &a[max_val(&a[0] , i + 1)]);
7 }
8
9 int max_val(char a [], int n)
10 {
11     int i, i_max = 0;
12
13     char massimo = a[0];
14
15     for (i = 1 ; i < n ; i++)
16
17         if (a[i] > massimo)
18         {
19             massimo = a[i];
20             i_max = i;
21         }
22
23     return i_max;
24 }
25
26 void scambiare_c(char *a, char *b)

```

```

28 {
    char temp;
30
    temp = *a;
32
    *a = *b;
34
    *b = temp;
36
}

```

Quesito n. 20

Scrivere una function C che dato un array di stringhe restituisce l'array ordinato in ordine crescente (usare l'algoritmo di **selezione di massimo**).

Riferimento

- Manuale di Programmazione in C
- Capitolo 5

Soluzione

```

0
void ord_sel_max_S(char elenco[][50], int n)
2 {
4     int i, j;
6     for (i = n - 1 ; i > 0; i--)
            scambiare_S(elenco[i], elenco[max_val_S(elenco, i + 1)]);
8 }

10
int max_val_S(char a[][50], int n)
12 {
14     int i, i_max = 0;
16     for (i = 1 ; i < n ; i++)
            if (strcmp(a[i], a[i_max]) > 0)
                i_max = i;
18
19     return i_max;
20 }

22
void scambiare_S (char *a, char *b)
24 {
26     char temp[100] = " ";
28     strcpy(temp, a);
30     strcpy(a, b);
32     strcpy(b, temp);
}

```

Quesito n. 21

Scrivere una function C che dato un array frastagliato di stringhe (array di puntatori a *char*) restituisce l'array ordinato in ordine crescente (usare l'algoritmo di **selezione di massimo**).

Riferimento

- Manuale di Programmazione in C
- Capitolo 5

Soluzione

```
0 void ord_sel_max_Sp(char *elenco[], int n)
2 {
4     int i;
4     for (i = n - 1 ; i > 0; i--)
6     {
6         scambiare_Sp(&elenco[i], &elenco[max_val_Sp(&elenco[0], i + 1)]);
8     }
8 }
10
12 int max_val_Sp(char *a[], int n)
14 {
14 int i, i_max = 0;
16     for (i = 1 ; i < n ; i++)
18         if (strcmp(a[i], a[i_max]) > 0)
20             i_max = i;
22 return i_max;
22 }
24 void scambiare_Sp(char **a, char **b)
26 {
26     char *temp;
28     temp = *a;
30     *a = *b;
32     *b = temp;
}
```

Quesito n. 22

Scrivere una function C che restituisce l'indice di una chiave contenuta in un array ordinato di caratteri (**ricerca binaria ricorsiva**).

Riferimento

- Manuale di Programmazione in C
- Capitolo 5

Soluzione

```
0
2 int ric_bin_ric(char chiave, char elenco[], int n)
3 {
4     int mediano;
6     if(n == 0)
8         return -1;
9     else
10        {
11            mediano = (n - 1)/2;
12
13            if(chiave == elenco[mediano])
14                return mediano;
15            else if (chiave < elenco[mediano])
16                return ric_bin_ric(chiave, elenco, mediano);
17            else
18                return mediano + 1 + ric_bin_ric(chiave, elenco + mediano + 1, n
19                    - (mediano + 1));
20        }
21 }
22 }
```

Quesito n. 23

Scrivere due function C per il calcolo del fattoriale di un numero naturale n , una che usa una tecnica iterativa e l'altra ricorsiva.

Riferimento

- Manuale di Programmazione in C
- Capitolo 5

Soluzione

```
0
2 int fattorialeI(int n)
3 {
4     int i, fatt = 1;
6     for(i = 1; i <= n; i++)
7         fatt = fatt * i;
8
9     return fatt;
10 }
```

```

10 }
12
13 int fattorialeI_ric(int n)
14 {
15     if ( n <= 1 )
16         // soluzione caso base
17         return 1;
18
19     else
20         // autoattivazione
21         return n * fattorialeI_ric(n - 1);
22 }

```

Quesito n. 24

Scrivere una function C per il calcolo della somma di un array di interi con approccio incrementale e tecnica ricorsiva.

Riferimento

- Manuale di Programmazione in C
- Capitolo 5

Soluzione

```

0
2 int sommaI_array_ric(int a[], int n)
3 {
4
5     if( n == 1)
6         return a[0];
7     else
8         return a[n - 1] + sommaI_array_ric(a, n - 1);
9
10 }

```

Quesito n. 25

Scrivere una function C per il calcolo della somma di un array di interi con un approccio divide et impera e tecnica ricorsiva.

Riferimento

- Manuale di Programmazione in C
- Capitolo 5

Soluzione

```
0
2 int sommaI_array_ric_DI(int a[], int primo, int ultimo)
3 {
4     int mediano;
5
6     if (primo == ultimo)
7         return a[primo];
8
9     else
10    {
11        mediano = (primo + ultimo)/2;
12
13        return sommaI_array_ric_DI(a, primo, mediano) + sommaI_array_ric_DI(a
14        , mediano + 1, ultimo);
15    }
16 }
```

Quesito n. 26

Scrivere una function C per il calcolo della somma di un array di interi con un approccio divide et impera e tecnica ricorsiva.

Riferimento

- Manuale di Programmazione in C
- Capitolo 5

Soluzione

```
0
2 int sommaI_array_ric_DI(int a[], int primo, int ultimo)
3 {
4     int mediano;
5
6     if (primo == ultimo)
7         return a[primo];
8
9     else
10    {
11        mediano = (primo + ultimo)/2;
12
13        return sommaI_array_ric_DI(a, primo, mediano) + sommaI_array_ric_DI(a
14        , mediano + 1, ultimo);
15    }
16 }
17
18 int sommaI_array_ric_DI_n(int a[], int n)
19 {
20     int mediano;
21
22     if (n == 1)
23         return a[0];
```

```

24 |     else
26 |     {
28 |         mediano = (n - 1)/2;
30 |
32 |         return sommaI_array_ric_DI_n(a + mediano + 1, n - (mediano + 1)) +
           sommaI_array_ric_DI_n(a, mediano + 1);
34 |     }
36 | }

```

Quesito n. 27

Scrivere una function C per il calcolo del massimo di un array di interi con un approccio incrementale e tecnica ricorsiva.

Riferimento

- Manuale di Programmazione in C
- Capitolo 5

Soluzione

```

0 |
2 | int MaxI_array_ric_mod(int a[], int n)
3 | {
4 |
6 |     if(n == 1)
7 |         return a[0];
8 |     else
9 |         return MAX_I(a[n-1], MaxI_array_ric_mod(a, n - 1));
10 | }
12 | int MAX_I(int a, int b)
13 | {
14 |
16 |     if(a > b)
17 |         return a;
18 |     else
19 |         return b;
20 | }

```

Quesito n. 28

Scrivere due function C per il calcolo del massimo di un array di interi con un approccio divide et impera e tecnica ricorsiva.

Riferimento

- Manuale di Programmazione in C
- Capitolo 5

Soluzione

```
0 int MAXI_array_ric_DI_n(int a[], int n)
2 {
4     int mediano;
6     if (n == 1)
8     {
10        return a[0];
12    }
14    else
16    {
18        mediano = (n - 1)/2;
20        return MAX_I(MAXI_array_ric_DI_n(a, mediano + 1),
22        MAXI_array_ric_DI_n(a + mediano + 1, n - (mediano + 1)));
24    }
26 }
28 int MAX_I(int a, int b)
30 {
32     if(a > b)
34     return a;
36     else
38     return b;
40 }
```

Quesito n. 29

Scrivere una function C per il calcolo dell' n -esimo numero della serie di Fibonacci ($f_n = f_{n-1} + f_{n-2} \forall n > 1$ e $f_0 = 0, f_1 = 1$). Scrivere inoltre una function per ridurre la complessità a $O(n)$.

Riferimento

- Manuale di Programmazione in C
- Capitolo 5

Soluzione


```

0 int fib_ric_el(int n)
2 {
4     //printf("n %d \n", n);
4     //getchar();

6     if (n <= 1 )
8         return n;

10    else return fib_ric_el(n - 1) + fib_ric_el(n - 2);
12 }

12 double fib_ric_PD(int n)
14 {

16     if (n <= 1 )
18         return (double)n;

20     if (fib_array[n] != 0.0)
22         return fib_array[n];
24     else
26     {
        fib_array[n] = fib_ric_PD(n-1) + fib_ric_PD(n-2);
        return fib_array[n];
    }
}

```

Quesito n. 30

Scrivere una function C che data la seguente *struct* aggiorna la carriera dello studente con un nuovo voto.

```

0 struct studente{
2     char *Cognome;
2     char *Nome;
4     int eta;
4     int libretto[18];
6 };
8 typedef struct studente Studente;

```

Riferimento

- Manuale di Programmazione in C
- Capitolo 5

Soluzione

```

0 Studente aggiorna_carriera(Studente stud, int pos_exam, int voto)
2 {

```

```

4     stud.libretto[pos_exam - 1] = voto;
6     return stud;
8 }
10 void aggiorna_carriera_p(Studente *stud, int pos_exam, int voto)
11 {
12     stud->libretto[pos_exam - 1] = voto;
13     //(*stud).libretto[pos_exam - 1] = voto;
14 }
16 // Nel main
17 Studente studente = {"Rossi", "Mario", 19, {28, 25, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}};
18 studente = aggiorna_carriera(studente, 3, 30);
20 // oppure
22 aggiorna_carriera_p(&studente, 3, 30);

```

Quesito n. 31

Scrivere una function C che dato un elenco di libri identificati dalla seguente *struct* restituisce l'indice del libro con costo massimo.

```

0 struct libro{
2     char *Titolo;
3     int costo ;
4 };
6 typedef struct libro Libro;

```

Riferimento

- Manuale di Programmazione in C
- Capitolo 5

Soluzione

```

0 int max_costo_libri(Libro scaffale[], int n)
1 {
2     int i, max, pos;
4     pos = 0;
6     for (i = 1 ; i < n; i++)
7         if (scaffale[pos].costo < scaffale[i].costo)
8             {
9                 pos = i;
10            }
12 }

```

```
14 }  
    return pos;
```

Quesito n. 32

Scrivere una una function C che genera k numeri pseudocasuali reali nell'intervallo $inf-sup$ dato dall'utente.

Riferimento

- Manuale di Programmazione in C
- Capitolo 5

Soluzione

```
0 void generazione_pseudocasuale(int k, int inf, int sup, float *pseudo)  
2 {  
4     srand(time(NULL));  
6     for(i = 0 ; i < k; i++)  
8         pseudo[i] = inf + (sup - inf) * (float)rand()/(float)RAND_MAX;  
}
```

Quesito n. 33

Scrivere un programma per creare un array di dimensione n in modo dinamico.

Riferimento

- Manuale di Programmazione in C
- Capitolo 5

Soluzione

```
0 #include <stdio.h>  
2 #include <malloc.h>  
4 int main() {  
6     char *array;  
8     printf("dammi la dimensione dell'array -> \n");  
9     scanf("%d", &n);  
10    array = (char *) malloc(n * sizeof(char));
```

```
12 | return 0;
14 | }
```

Quesito n. 34

Scrivere un programma che alloca dinamicamente un array di interi mediante la function *calloc*, permette di inserire gli elementi all'utente e liberi successivamente la memoria allocata.

Riferimento

- Manuale di Programmazione in C
- Capitolo 5

Soluzione

```
0 | #include <stdio.h>
2 | #include <malloc.h>
4 | int main() {
6 |     char *array;
8 |     printf("dammi la dimensione dell'array -> \n");
   |     scanf("%d", &n);
10 |     array = (int *) calloc(n, sizeof(int));
12 |     if (array == NULL)
14 |     {
   |         printf("Memoria insufficiente \n");
16 |         exit(1);
   |     }
18 |     for(i= 0; i < n ; i++)
20 |     {
   |         printf("Dammi %d-esimo elemento: \n", i+1);
22 |         scanf("%d", &array[i]);
24 |     }
26 |     for(i= 0; i < n ; i++)
   |     {
28 |         printf("L' %d-esimo elemento e': %d \n", i+1, array[i]);
   |     }
30 |     free(array);
32 |     return 0;
34 | }
```

Testi consigliati

I testi consigliati sono

- A. Ciaramella, G. Giunta – Manuale di programmazione in C. Scaricabile dalla piattaforma di e-learning della Facoltà di Scienze e Tecnologie
- A. Kelley, I. Pohl – C: didattica e programmazione. Pearson Education Italia. 2018

Note sui quesiti

I quesiti sono tratti da “Manuale di Programmazione in C” di Angelo Ciaramella e Giulio Giunta. I quesiti sono classificati secondo un grado crescente di complessità, finalizzati alla simulazione da parte dello studente iscritto al corso di una verifica intercorso e/o di profitto finale con le relative risposte descritte e commentate, indicando altresì nel dettaglio le pagine del testo di riferimento da consultare del manuale di riferimento.