

# Informática-Ingeniería en Mecatrónica-2022

## Trabajo Práctico N° 3 - A

### Parte 1 – Estructuras de datos y cadenas

Borquez Juan. Legajo:13567

1. Escriba un programa que:

- Genere un vector de tamaño N con valores aleatorios enteros, utilizando la función rand(). Use la función srand() para obtener una mejor aleatoriedad.
- Recorra el vector y guarde el promedio de los valores en una variable
- Encuentre el menor de los elementos

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <time.h>
4
5  int main(void) {
6      short int max = 100, maxval = 10, N;
7      int vector[max], minimo = maxval + 1;
8      float promedio = 0;
9
10     srand(time(NULL));
11     printf("TP3-P1-EJ1:\n\n");
12     printf("El tamaño del vector es %d. A continuacion indique el tamaño N a utilizar: ", max);
13     scanf("%hd", &N);
14     srand(time(NULL));
15     for (short int i = 0; (i < N) && (i < max); i++){
16         vector[i] = rand()%(maxval + 1);
17         promedio += vector[i];
18         if (vector[i] < minimo){
19             minimo = vector[i];
20         }
21         printf("VECTOR[%hd]: %d\n", i, vector[i]);
22     }
23     for (short int i = 0; (i < N) && (i < max); i++){
24         if (i == 0){
25             minimo = vector[i];
26         }
27         else{
28             if (vector[i] < minimo){
29                 minimo = vector[i];
30             }
31         }
32     }
33     promedio/=(float)N;
34     printf("\nEL promedio de los numeros del vector es: %f", promedio);
35     printf("\nEl valor minimo es: %d\n", minimo);
36     return 0;
```

## Informática-Ingeniería en Mecatrónica-2022

### Trabajo Práctico N° 3 - A

38 }

<https://replit.com/@JuanBorquez/TP3-P1-EJ1#main.c>

2. Implementar una función que genere una matriz con todos los elementos en 0 excepto aquellos para los que  $i+j$  sea par; para estos elementos generar un valor aleatorio entre 1 y 10000.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <time.h>
4  #define size 10
5
6  void cargar(int matriz[0][size], short int f, short int c);
7  void imprimir(int matriz[0][size], short int f, short int c);
8
9  int main(void) {
10     int matriz[size][size];
11     short int f, c;
12     printf("TP3-PARTE1-EJERCICIO 2");
13     printf("\n\nLa matriz es de numeros enteros de %d por %d", size, size);
14     printf("\nA continuacion indique las cantidad de filas f: "); scanf("%hd", &f);
15     printf("A continuacion indique las cantidad de columnas c: "); scanf("%hd", &c);
16     cargar(matriz, f, c);
17     imprimir(matriz, f, c);
18     return 0;
19 }
20 void cargar(int matriz[][size], short int f, short int c){
21     srand(time(NULL));
22     for (short int j = 0; (j < c) && (j < size); j++){
23         for (short int i = 0; (i < f) && (i < size); i++){
24             matriz[i][j] = ((i + j)%2)? 0:(1 + rand()%1000);
25         }
26     }
27 }
28 void imprimir(int matriz[0][size], short int f, short int c){
29     for (short int i = 0; (i < f) && (i < 10); i++){
30         printf("\n");
31         for (short int j = 0; (j < c) && (j < 10); j++){
32             printf("\t\t[%hd][%hd]: %d", i, j, matriz[i][j]);
33         }
34     }
35 }
```

<https://replit.com/@JuanBorquez/TP3-P1-EJ2#main.c>

3. Escriba un programa que lea una cadena con una operación matemática básica de dos términos (suma, resta, multiplicación y división) y calcule el resultado. Por ej. el

# Informática-Ingeniería en Mecatrónica-2022

## Trabajo Práctico N° 3 - A

usuario ingresa "2.4+3", el resultado a devolver es "5.4". Considere los datos de entrada números flotantes. Considere una cadena máxima de 100 caracteres.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <ctype.h>
4  #include <string.h>
5
6  int main(void){
7      float op1, op2, resultado;
8      char operador;
9      printf("TP3-PARTE1-EJERCICIO 3\n");
10     printf("\nA continuacion indicar una operacion aritmetica basica (+,-,/,*) con 2
11     operandos racionales:\n\n");
12     printf("\t\tPor ejemplo: 2.3+5 (observar que no se colocan espacios)\n");
13     printf("\nLa cantidad maxima de caracteres a ingresar es de 100\n");
14     printf("\n\tOPERACION:\n");
15     scanf("%f%c%f", &op1, &operador, &op2);
16     /*printf("\nop1: %f", op1);
17     printf("\nop2: %f", op2);
18     printf("\noperador: %c", operador);*/
19     switch (operador){
20         case '+':
21             resultado = op1 + op2;
22             printf("\nResultado: %f", resultado);break;
23         case '-':
24             resultado = op1 - op2;
25             printf("\nResultado: %f", resultado);break;
26         case '/':
27             resultado = op1/op2;
28             printf("\nResultado: %f", resultado);break;
29         case '*':
30             resultado = op1*op2;
31             printf("\nResultado: %f", resultado);break;
32         default:
33             printf("No se ingreso un caracter de operación valido");
34     }
}
```

<https://replit.com/@JuanBorquez/TP3-P1-EJ3#main.c>

4. Escriba un programa que lea una frase y evalúe en forma recursiva si la misma corresponde a un palíndromo. Considere una cadena máxima de 100 caracteres.

# Informática-Ingeniería en Mecatrónica-2022

## Trabajo Práctico N° 3 - A

```
1  #include<stdio.h>
2  #include<stdlib.h>
3  #include<string.h>
4
5  /*La siguiente es una función recursiva que determina si una cadena de caracteres es palíndromo
6
7  -En el primer llamado recibe un puntero a la cadena de caracteres (word) de origen y el tamaño de la misma
8  -En los llamados recursivos el puntero word es en realidad un puntero a una posición en la cadena de caracteres de
9  origen
10 De este modo no hay que crear varias cadenas. Es decir que siempre se trabaja con la cadena de origen
11 - size_of_word es el tamaño de la cadena de origen cuando se llama la función por primera vez
12 - size_of_word se interpreta como la longitud de la porción de la cadena de origen que se toma en
13 cada llamado recursivo a la función.
14 - Si la cadena es un palíndromo devuelve 1 y devuelve 0 si no lo es*/
15 short int is_palindrome(char *word, short int size_of_word);
16 void no_space(char* word);
17
18 int main(void){
19     char word[100];
20     printf("TP3-PARTE1-EJERCICIO 4\n\n");
21     printf("A continuacion indique una palabra de no mas de 100 caracteres:\n");
22     printf("\nPALABRA: \t");
23     scanf("%[^\\n]", word);
24     no_space(word);
25     if(is_palindrome(word, strlen(word))){
26         printf("\nLa frase SI es un palindromo\n");
27     }
28     else{
29         printf("\nLa palabra NO es un palindromo\n");
30     }
31 }
32 short int is_palindrome(char *word, short int size_of_word){
33     if (size_of_word > 1){
34         //Si el tamaño de la porción considerada es mayor a 1 hacemos otra determinacion
35         if(word[0] == word[size_of_word-1]){//compara el pultimo y primer carácter de la porción
36             considerada
37                 //si son iguales hace otro llamado a la función
38                 return(is_palindrome(word + 1, size_of_word - 2));
39                 //word+1 será un puntero al caracter posterior al primero en la porción de cadena
40             considerada
41                 //Se restan dos caracteres a la porción en size_of_word - 2;
42         }
43         else{
44             //si en cierto momento el caracter inicial y final no coinciden entonces no es palindromo
45             return 0;
46         }
47     }
48     else{
49         //si el tamaño de la porción de cadena considerada es 0 o 1 significa que la palabra sí es palindromo
50         //Esto incluso cuando la cadena de origen tiene un solo caracter o ninguno
51         return 1;
52     }
53 }
54 void no_space(char* word){
```

## Informática-Ingeniería en Mecatrónica-2022

### Trabajo Práctico N° 3 - A

```
55 short int i = 0, k = 0;
56 while(word[i] != '\0'){
57     if (word[i] != ' '){
58         if (k > 0){
59             word[i-k] = word[i];
60         }
61     }
62     else{
63         k++;
64     }
65     i++;
66 }
67 if(k > 0){
    word[i-k] = '\0';
}
}
```

<https://replit.com/@JuanBorquez/TP3-P1-EJ4#main.c>

5. Modifique el ejercicio 12 del práctico anterior (el de las fechas) utilizando un struct que represente una fecha completa. Realice al menos una función que reciba por parámetro una fecha. Defina la estructura fecha como un tipo definido por el usuario (typedef)

## Informática-Ingeniería en Mecatrónica-2022

### Trabajo Práctico Nº 3 - A

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  typedef struct{
5      short int DD;
6      short int MM;
7      short int AAAA;
8  } date;
9  //devuelve la fecha anterior
10 date previous(date actual);
11
12 //devuelve la fecha posterior
13 date next(date actual);
14
15 short int is_leap(date actual);
16 short int days_month(date actual);
17 short int is_valid(date actual);
18
19 int main(void){
20     date actual, prev, nex;
21     char c;
22     short int daysMM;
23     printf("TP3-PARTE1-EJERCICIO 5");
24     printf("\n\nA continuacion indicar una fecha con el formato DD/MM/AAAA\n");
25     printf("\nSi indica una fecha incorrecta se volvera a pedir la fecha\n");
26     printf("Presione una tecla: "); c = getchar();
27     do{
28         system("clear");
29         do{
30             system("clear");
31             printf("\nDD:\t");scanf("%2hd", &actual.DD);
32         }while((actual.DD < 1)|| (actual.DD > 31));
33         do{
34             system("clear");
35             printf("\nMM:\t");scanf("%2hd", &actual.MM);
36         }while((actual.MM < 1)|| (actual.MM > 12));
37         system("clear");
38         printf("\nAAAA:\t");scanf("%4hd", &actual.AAAA);
39     }while(!is_valid(actual));
40     system("clear");
41     prev = previous(actual);
42     nex = next(actual);
43     daysMM = days_month(actual);
44
45     printf("\nFecha actual: \t%hd/%hd/%hd", actual.DD, actual.MM, actual.AAAA);
46     printf("\nFecha anterior: \t%hd/%hd/%hd", prev.DD, prev.MM, prev.AAAA);
47     printf("\nFecha siguiente: \t%hd/%hd/%hd", nex.DD, nex.MM, nex.AAAA);
48     printf("\nUltimo dia del mes: \t%hd/%hd/%hd", daysMM, actual.MM, actual.AAAA);
49     printf("\nDias para que termine el mes: \t%d\n\n", daysMM-actual.DD);
```

## Informática-Ingeniería en Mecatrónica-2022

### Trabajo Práctico N° 3 - A

```
50
51 }
52 date previous(date actual){
53     date previous;
54     if((actual.DD - 1) != 0){
55         previous.DD = actual.DD-1;
56         previous.MM = actual.MM;
57         previous.AAAA = actual.AAAA;
58     }
59     else{
60         previous.MM = actual.MM - 1; previous.AAAA = actual.AAAA;
61         previous.DD = days_month(previous);
62         previous.MM = ((actual.MM-1) != 0)? actual.MM - 1:12;
63         previous.AAAA = ((actual.MM-1) != 0)? actual.AAAA:actual.AAAA-1;
64     }
65     return (previous);
66 }
67 date next(date actual){
68     date next;
69     if(actual.DD < days_month(actual)){
70         next.DD = actual.DD + 1;
71         next.MM = actual.MM;
72         next.AAAA = actual.AAAA;
73     }
74     else{
75         next.DD = 1;
76         next.MM = (actual.MM != 12)? actual.MM + 1:1;
77         next.AAAA = (actual.MM != 12)? actual.AAAA:actual.AAAA+1;
78     }
79     return (next);
80 }
81 short int days_month(date actual){
82     if (actual.MM != 2){
83         return((((actual.MM%2)&&(actual.MM <= 7))||(!(actual.MM%2)&&(actual.MM >
84 7))||((actual.MM == 0)? 31:30));
85     }
86     else{
87         return((is_leap(actual))?29:28);
88     }
89 }
90 short int is_leap(date actual){
91     // es bisiesto si es divisible por 4 y no por 100 o si es divisible por 400
92     return((((!(actual.AAAA%4)&&(actual.AAAA%100))||!(actual.AAAA%400))? 1:0);
93 }
94 short int is_valid(date actual){
95     return((actual.DD <= days_month(actual))? 1:0);
96 }
```

6. Escriba un programa que calcule el producto vectorial de 2 vectores en R3 utilizando structs para representar los vectores

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <time.h>
4
5  typedef struct{
6      float xx;
7      float yy;
8      float zz;
9  }vector;
10
11 //calcula el producto cruz de dos vectores en R3
12 vector cruz(vector u, vector v);
13
14 int main(void) {
15     vector u,v,w;
16     printf("TP3-P1-EJERCICIO6");
17     printf("\nIndique las componentes del primer vector: ");
18     printf("\ntu_xx = ");scanf("%f", &u.xx);
19     printf("\tu_yy = ");scanf("%f", &u.yy);
20     printf("\tu_zz = ");scanf("%f", &u.zz);
21     printf("\nIndique las componentes del segundo vector: ");
22     printf("\ntv_xx = ");scanf("%f", &v.xx);
23     printf("\tv_yy = ");scanf("%f", &v.yy);
24     printf("\tv_zz = ");scanf("%f", &v.zz);
25     w = cruz(u, v);
26     system("clear");
27     printf("\nu = [%f, %f, %f]", u.xx, u.yy, u.zz);
28     printf("\nv = [%f, %f, %f]", v.xx, v.yy, v.zz);
29     printf("\n\nW = u X v = [%f, %f, %f]\n", w.xx, w.yy, w.zz);
30     return 0;
31 }
32 vector cruz(vector u, vector v){
33     vector w;
34     w.xx = u.yy*v.zz - v.yy*u.zz;
35     w.yy = -u.xx*v.zz + v.xx*u.zz;
36     w.zz = u.xx*v.yy - v.xx*u.yy;
37     return w;
38 }

```