

### **PARTE 3 – Subrutinas**

Los ejercicios planteados en esta sección deben utilizar de forma obligatoria subrutinas para su resolución.

12. Escribir un programa que le pida al usuario una fecha del estilo DD/MM/AAAA y determine

- a) El día anterior y posterior.
- b) El último día del mes y cuantos días faltan para el mismo.

Considere la existencia de los años bisiestos.

```

1  #include <stdio.h>
2  #include <string.h>
3  #include <stdlib.h>
4  #include <string.h>
5
6
7  //determina el número de día en la semana de la fecha indicada. 1 es lunes, 2 es martes...
8  short int date_number(short int DD, short int MM, short int AAAA);
9  //determina el número de días en el mes MM del año AAAA
10 short int month_days(short int MM, short int AAAA);
11 //determina si el año AAAA es bisiesto. Devuelve 1 si es y 0 si no
12 short int is_leap(short int AAAA);
13 //indica el nombre del día :day_number
14 //recibe números del -7 al 7
15 char *day_name(short int day_number);
16
17
18 int main(void) {
19     short int DD, MM, AAAA, var_date_number, var_month_days;
20     char* var_day_name;
21
22     printf("\n\nTP2-P2-EJ12\n\n");
23
24     printf("A continuacion indique una fecha del tipo DD/MM/AAAA: \n");
25
26     printf("\tDD: "); scanf("%hd", &DD);
27     printf("\tMM: "); scanf("%hd", &MM);
28     printf("\tAAAA: "); scanf("%hd", &AAAA);
29
30     var_date_number = date_number(DD,MM,AAAA);
31
32
33     var_day_name = day_name(var_date_number - 1);
34
35     printf("\n\nEl día anterior es: ");puts(var_day_name);
36     free(var_day_name);
37
38     var_day_name = day_name(var_date_number + 1);
39     printf("El día siguiente es: ");puts(var_day_name);
40     free(var_day_name);
41
42     var_month_days = month_days(MM, AAAA);
43     var_date_number += (var_month_days - DD);
44
45     var_day_name = day_name(var_date_number);
46     printf("El último día del mes es: ");puts(var_day_name);
47     printf("La cantidad de días que faltan para el mismo es: %d\n\n", var_month_days-DD);
48 }
49

```

```

50 short int date_number(short int DD, short int MM, short int AAAA){
51     /* 1/01/2022 ES SABADO */;
52     short int date_number; //numero de día
53     //cantidad de días en el mismo año AAAA hasta la fecha; el menos 1 es por el primero de
54     enero
55     short int days_to_date = -1;
56     //numero de años equivalentes desde 2022 hasta AAAA
57     int years_to_date;
58
59     //Lo siguiente es equivalente a contar los años y sumar la cantidad de bisiestos
60     years_to_date = ((AAAA-2022)*5)/4;
61
62     //Suma un año mas si el año es posterior a 2022 y uno de los que sobra es 2024 que es
63     bisiesto
64     //eso solo ocurre cuando el resto de la división es solo 3
65     //Si el año es posterior pero en el resto no está el 2024 entonces no suma nada
66     years_to_date += ((AAAA-2022)%4 == 3)? 1:0;
67
68     //Resta un año mas si el año es anterior a 2022 y el resto contiene a 2020 que es bisiesto
69     //eso solo ocurre cuando el resto de la división es solo -3 o -2
70     //si el año es anterior pero en el resto no está el 2020 entonces no suma nada
71     years_to_date -= (((AAAA-2022)%4 == -2)||((AAAA-2022)%4 == -3))? 1:0;
72
73     for (short int i = 1; i < MM; i++){
74         days_to_date += month_days(i, AAAA);
75     }
76     days_to_date += DD;
77
78     date_number = (6 + days_to_date%7 + years_to_date%7)%7;
79     return date_number;
80 }
81 short int month_days(short int MM, short int AAAA){
82     short int month_days;
83     if(((MM % 2) && (MM < 8))||(!(MM % 2)&&(MM >= 8))){
84         month_days = 31;
85     }
86     else{
87         if (MM != 2){
88             month_days = 30;
89         }
90         else{
91             month_days = (is_leap(AAAA))? 29:28;
92         }
93     }
94     return month_days;
95 }
96 short int is_leap(short int AAAA){
97     short int is_leap = ((!(AAAA%4)&&(AAAA%100))||!(AAAA%400))? 1:0;
98     return is_leap;

```

```

99     }
100     char *day_name(short int day_number){
101         //tomamos el modulo 7 nuevamente en caso de que se ingrese 7 o -7 sino no hace falta
102         char* day_name = (char*)malloc(10*sizeof(char));
103         day_number = (day_number >= 0)? (day_number%7):(7 + day_number);
104         switch(day_number){
105             case 0:
106                 strcpy(day_name, "DOMINGO");break;
107             case 1:
108                 strcpy(day_name, "LUNES");break;
109             case 2:
110                 strcpy(day_name, "MARTES");break;
111             case 3:
112                 strcpy(day_name, "MIERCOLES");break;
113             case 4:
114                 strcpy(day_name, "JUEVES");break;
115             case 5:
116                 strcpy(day_name, "VIERNES");break;
117             case 6:
118                 strcpy(day_name, "SABADO");break;
119         }
120         return(day_name);
121     }

```

<https://replit.com/@JuanBorquez/TP2-P3-EJ12#main.c>

13. Escribir un programa que permita ingresar por teclado un valor inicial a, un valor final b, y un número c, y que cuente la cantidad de números divisibles por c que hay en el rango [a,b].

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  short int num_multiplos(float min, float max, float divisor);
5
6  int main(void) {
7      float a, b, c;
8      printf("\nTP2-P3-EJ13: \n\n");
9
10     printf("Determina la cantidad de numeros divisibles por un numero c en el intervalo [a,
11 b]\n\n");
12
13     printf("Indicar el primer numero a(minimo) = "); scanf("%f", &a);
14     printf("Indicar el segundo numero b(maximo) = "); scanf("%f", &b);
15     while (b < a){
16         printf("\nEl numero b ha de ser mayor o igual al numero a");
17         printf("\nIndicar el segundo numero b(maximo) = "); scanf("%f", &b);
18         system("clear");
19     }
20     printf("Indicar el tercer numero c(divisor) = "); scanf("%f", &c);
21     printf("\nEl numero de multiplos de c en [a, b] es = %d\n", num_multiplos(a, b, c));
22 }
23 short int num_multiplos(float min, float max, float divisor){
24     short int i = (int)min/divisor;
25     short int k = (i*divisor == min)? 1:0;
26     i++;
27     while((divisor*i) <= max){
28         k++;
29         i++;
30     }
31     return k;
32 }

```

<https://replit.com/@JuanBorquez/TPS-P3-EJ13#main.c>

14. Escribir un programa que calcule las siguientes dos ecuaciones:

$$\sum_{i=0}^{100} \sum_{j=0}^{100} \frac{i+j}{i-j} \forall i, \frac{j}{(i-j)} \neq 0 \quad \sum_{i=0}^{100} \sum_{j=0}^{100} \frac{i*j}{i+j} \forall i, \frac{j}{i+j} \text{son números pares}$$

```

1  #include <stdio.h>
2  #include <math.h>
3
4  //en esta primera hacemos el calculo directo
5  float primer_ec(void);
6
7  //en esta otra hscemos el calculo por recursión
8  float sum_in(short int nj, short int i);
9  float sum_out(short int ni);
10
11 int main(void) {
12     printf("TP2-P3-EJ14: \n");
13     printf("\nEl resultado de la primera sumatoria es: %f", primer_ec());
14     printf("\nEl resultado de la segunda sumatoria es: %f\n", sum_out(100));
15     return 0;
16 }
17 float primer_ec(void){
18     float sum = 0;
19     for (short int i = 0; i <= 100; i++){
20         for (short int j = 0; j <= 100; j++){
21             if ((j != 0)&&((i-j) != 0)){
22                 sum += (float)(i + j)/(i - j);
23             }
24         }
25     }
26     return sum;
27 }
28 float sum_in(short int nj, short int i){
29     if (nj > 0){
30         if (!(nj % 2)){
31             return((float)nj*i/(nj+i) + sum_in(nj-1, i));
32         }
33         else{
34             return(sum_in(nj-1, i));
35         }
36     }
37     else{
38         return 0;
39     }
40 }
41 float sum_out(short int ni){
42     if(ni > 0){
43         if(!(ni%2)){
44             return(sum_in(100, ni) + sum_out(ni-1));
45         }
46         else{
47             //dismiss las impares
48             return(sum_out(ni-1));
49         }
50     }
51     else{
52         return(0);
53     }
54 }

```

<https://replit.com/@JuanBorquez/TP2-P3-EJ14#main.c>

15. Implemente una función que calcule el factorial de un número mediante recursión devuelva dicho factorial. Llame a esa función desde el procedimiento main y muestre el resultado por pantalla.

```
1  #include <stdio.h>
2
3  int factorial(int num);
4
5  int main(void) {
6      int N;
7      printf("TP2-P3-EJ15: \n");
8      printf("Programa que calcula el factorial de un numero N por recursion\n\n");
9      printf("Indique un numero entero N = "); scanf("%d", &N);
10
11     printf("\n\nEl factorial del numero N es: %d\n", factorial(N));
12     return 0;
13 }
14
15 int factorial(int num){
16     if (num > 0){
17         return(num*factorial(num-1));
18     }
19     return 1;
20 }
```

<https://replit.com/@JuanBorquez/TP2-P3-EJ15#main.c>

16. Implemente recursivamente el cálculo de la sucesión de Fibonacci:

$f_0 = 0$

$f_1 = 1$

$f_n = f_{n-1} + f_{n-2} \quad \forall n \geq 2$

```
1  #include <stdio.h>
2
3  int fibonacci(int n);
4
5  int main(void) {
6      int n;
7      printf("TP2-P3-EJ16\n");
8      printf("Programa que determina el término n-simo de la serie de Fibonacci\n");
9      printf("\nIndique el índice n = ");scanf("%d", &n);
10     printf("\nEl termino %d-simo es = %d \n", n, fibonacci(n));
11     return 0;
12 }
13 int fibonacci(int n){
14     if (n >= 2){
15         return(fibonacci(n-2) + fibonacci(n-1));
16     }
17     else{
18         return((n == 1)? 1:0);
19     }
20 }
```

<https://replit.com/@JuanBorquez/TP2-P3-EJ16#main.c>

17. Escriba un programa que lea por teclado 3 coeficientes a, b y c, un valor inicial x1, un valor final x2 y un incremento delta, y que calcule p(x) utilizando la función que se muestra a continuación, en el intervalo [x1, x2], según el incremento delta. Muestre los

valores calculados por pantalla.  $p(x) = ax^2 + bx + c$



```

1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <math.h>
4
5 void cuadratica(float a, float b, float c, float x1, float x2, float delta);
6
7 int main(void) {
8     float a, b, c, x1, x2, delta;
9     printf("TP2-P3-EJ17");
10    printf("\n\nPrograma que calcula los valores de la funcion cuadratica siguiente\n");
11    printf("p(x) = ax^2 + bx + c\n");
12
13    printf("\nA continuacion indicar los coeficientes: ");
14    printf("\n\ta: "); scanf("%f", &a);
15    printf("\tb: "); scanf("%f", &b);
16    printf("\tc: "); scanf("%f", &c);
17
18    printf("\nA continuacion indicar un primer valor x1, un valor final x2 y un deltax: \n");
19    printf("\tx1 = "); scanf("%f", &x1);
20    printf("\tx2 = "); scanf("%f", &x2);
21
22    while (x2 < x1){
23        printf("\nDebe ser x2 > x1\n");
24        printf("\tx2 = "); scanf("%f", &x2);
25        system("clear");
26    }
27    printf("\tdelta = "); scanf("%f", &delta);
28    while (delta <= 0){
29        printf("\nDebe ser deltax > 0\n");
30        printf("\tdelta = "); scanf("%f", &delta);
31        system("clear");
32    }
33    system("clear");
34    cuadratica(a,b,c,x1,x2,delta);
35    return 0;
36 }
37 void cuadratica(float a, float b, float c, float x1, float x2, float delta){
38     float x = x1;
39     printf("x\t\t\t\t\t p(x)\n");
40     printf("%f\t\t\t\t\t %f", x, a*pow(x, 2) + b*x + c);
41     while ((x += delta) < x2){
42         printf("\n%f\t\t\t\t\t %f", x,a*pow(x, 2) + b*x + c);
43     }
44     printf("\n%f\t\t\t\t\t %f\n", x2 ,a*pow(x2, 2) + b*x2 + c);
45 }

```