

```

#include <stdio.h>

// Variable declaration:
extern int a, b;
extern int c;
extern float f;

int main () {

    /* variable definition: */
    int a, b;
    int c;
    float f;

    /* actual initialization */
    a = 10;
    b = 20;

    c = a + b;
    printf("value of c : %d \n", c);

    f = 70.0/3.0;
    printf("value of f : %f \n", f);

    return 0;
}

```

Imagen 1. Declaración e inicialización de variables.

```

#include <stdio.h>
#include <stdlib.h>
int main()
{
    char card_name[3];
    puts("Enter the card_name: ");
    scanf("%2s", card_name);
    int val = 0;
    if (card_name[0] == 'K') {
        val = 10;
    } else if (card_name[0] == 'Q') {
        val = 10;
    } else if (card_name[0] == 'J') {
        val = 10;
    } else if (card_name[0] == 'A') {
        val = 11;
    } else {
        val = atoi(card_name);
    }
    printf("The card value is: %i\n", val);
    return 0;
}

```

Imagen 2. Ciclo if-else.

```

#include <stdio.h>
void go_south_east(int lat, int lon)
{
    lat = lat - 1;
    lon = lon + 1;
}

int main()
{
    int latitude = 32;
    int longitude = -64;
    go_south_east(latitude, longitude);
    printf("Avast! Now at: [%i, %i]\n", latitude, longitude);
    return 0;
}

```

Pass in the latitude and longitude.

Decrease the latitude.

Increase the longitude.

Imagen 3. Funciones.

```

#include <stdio.h>

float promedio(int valores[], int cantidad) {
    int i;
    float suma = 0.0;

    for (i = 0; i < cantidad; ++i)
        suma += valores[i];

    return suma / (float) cantidad;
}

int main() {
    int notas[10];
    char nombre[20];
    char opcion[3];
    int n, i;

    do {
        printf("Ingrese nombre del alumno: ");
        scanf("%s", nombre);

        printf("Cuantas notas tiene %s? ", nombre);
        scanf("%d", &n);

        for (i = 0; i < n; ++i) {
            printf("Nota %d: ", i + 1);
            scanf("%d", &notas[i]);
        }

        printf("El promedio de %s es %.1f\n", nombre, promedio(notas, n));

        printf("Desea calcular mas promedios (si/no)? ");
        scanf("%s", opcion);

    } while (opcion[0] == 's' || opcion[0] == 'S');

    return 0;
}

```

Imagen 4. Funcion y ciclos.

```

#include <stdio.h>

int main () {

    int var = 20;    /* actual variable declaration */
    int *ip;         /* pointer variable declaration */

    ip = &var; /* store address of var in pointer variable*/

    printf("Address of var variable: %x\n", &var );

    /* address stored in pointer variable */
    printf("Address stored in ip variable: %x\n", ip );

    /* access the value using the pointer */
    printf("Value of *ip variable: %d\n", *ip );

    return 0;
}

```

Live Demo

Imagen 5. Apuntadores.

```

8  #define ALTURA 3
9  #define ANCHURA 3
10
11 #include <stdio.h> // Para printf
12
13 int main() {
14     int matriz[ALTURA][ANCHURA] = {
15         {20, 50, 80},
16         {500, 12, 44},
17         {56, 4, 3},
18     };
19     // Iniciar los números, suponiendo que son los primeros
20     int menor = matriz[0][0];
21     int mayor = matriz[0][0];
22     // Recorrer la matriz y comparar
23     for (int y = 0; y < ALTURA; y++) {
24         for (int x = 0; x < ANCHURA; x++) {
25             int elementoActual = matriz[y][x];
26             if (elementoActual > mayor) mayor = elementoActual;
27             if (elementoActual < menor) menor = elementoActual;
28         }
29     }
30     printf("Mayor: %d\n", mayor);
31     printf("Menor: %d\n", menor);
32     return 0;
33 }

```

Imagen 6. Matrices.

```

01 #include <stdio.h>
02
03 void qs(int lista[],int limite_izq,int limite_der)
04 {
05     int izq,der,temporal,pivote;
06
07     izq=limite_izq;
08     der = limite_der;
09     pivote = lista[(izq+der)/2];
10
11     do{
12         while(lista[izq]<pivote && izq<limite_der)izq++;
13         while(pivote<lista[der] && der > limite_izq)der--;
14         if(izq <=der)
15         {
16             temporal= lista[izq];
17             lista[izq]=lista[der];
18             lista[der]=temporal;
19             izq++;
20             der--;
21         }
22     }
23
24     while(izq<=der);
25     if(limite_izq<der){qs(lista,limite_izq,der);}
26     if(limite_der>izq){qs(lista,izq,limite_der);}
27
28 }
29
30 void quicksort(int lista[],int n)
31 {
32     qs(lista,0,n-1);
33 }
34
35 int main(int argc, const char * argv[])
36 {
37
38     int lista[] = {100,56,0,1,-45,2,46,5,9,6,67,23,5};
39     int size = sizeof(lista)/sizeof(int);
40
41     printf("Lista Desordenada \n");
42
43     for (int i=0; i<size; i++) {
44         printf("%d",lista[i]);
45         if(i<size-1)
46             printf(",");
47     }
48
49     printf("\n");
50     quicksort(lista,size);
51
52     printf("Lista Ordenada \n");
53     for (int i=0; i<size; i++) {
54         printf("%d",lista[i]);
55         if(i<size-1)
56             printf(",");
57     }
58
59     return 0;
60 }

```

Imagen 7. Algoritmo de ordenamiento Quicksort.

Bibliografía:

[Imagen 1]: Tutorialspoint.com. *C - Variables - Tutorialspoint*. [Disponible en línea]: https://www.tutorialspoint.com/cprogramming/c_variables.htm [Acceso 28 Agosto 2020].

[Imagen 2]: Griffiths, D. and Griffiths, D., 2012. *Head First C*. Beijing: O'Reilly, p.15. [Disponible en línea]: <http://karadev.net/uroci/filespdf/files/head-first-c-o-reilly-david-griffiths-dawn-griffiths.pdf>

[Imagen 3]: Griffiths, D. and Griffiths, D., 2012. *Head First C*. Beijing: O'Reilly, p.15. [Disponible en línea]: <http://karadev.net/uroci/filespdf/files/head-first-c-o-reilly-david-griffiths-dawn-griffiths.pdf>

[Imagen 4]: Progra.usm.cl. *Promedios De Alumnos — Programación*. [Disponible en línea]: <http://progra.usm.cl/apunte/c/promedios-alumnos.html> [Acceso 28 Agosto 2020].

[Imagen 5]: Tutorialspoint.com. *C - Variables - Tutorialspoint*. [Disponible en línea]: https://www.tutorialspoint.com/cprogramming/c_pointers.htm [Acceso 28 Agosto 2020].

[Imagen 6]: Número mayor y menor de matriz en C - Parzibyte's blog [Disponible en línea]: <https://parzibyte.me/blog/2019/10/28/numero-mayor-menor-matriz-c/> [Acceso 28 Agosto 2020].

[Imagen 7]: Quicksort en C – Algoritmo de ordenamiento | Codigoprogramacion. [Disponible en línea]: <http://codigoprogramacion.com/cursos/tutoriales-c/quicksort-en-c-algoritmo-de-ordenamiento.html#.X0hU7X61vIU> [Acceso 28 Agosto 2020].