



UNIVERSIDAD AUTÓNOMA DE QUERÉTARO
FACULTAD DE INGENIERÍA

Ingeniería en Automatización

Profesor: Sergio Miguel Delfín Prieto

Programacion Grupo 14

2° Semestre

Tarea 10

Diego Joel Zuñiga Fragoso

Exp: 317684

Querétaro, Qro. a 27/04/2023

1. Algoritmo 1: búsqueda del valor mínimo

```
#include <stdio.h>
int main()
{
    int T,*minIndex,*j,*i,*t;

    do
    {
        printf("Ingrese el tamaño del vector del que se sacara el valor minimo: ");
        scanf("%d",&T);
    }
    while (T<2);

    int A[T];

    for(int k=0;k<T;k++)
    {
        printf("\nIngrese el valor %d del arreglo: ",k+1);
        scanf("%d",&A[k]);
    }

    i=&A[0];
    t=&A[T-1];
    minIndex=&A[0];

    for(i;i<t;i++)
        for(j=i+1;j<t+1;j++)
        {
            if(*minIndex>*j)
                minIndex=j;
        }
    printf("\nEl valor minimo es %d",*minIndex);

    return 0;
}
```

2. Algoritmo 2: ordenar por selección

```
#include <stdio.h>
int main()
{
    int T,*minIndex,*j,*i,*t,res;

    do
    {
```

```

        printf("Ingrese el tamaño del vector del que se sacara el valor
minimo: ");
        scanf("%d",&T);
    }
    while (T<2);

    int A[T];

    for(int k=0;k<T;k++)
    {
        printf("\nIngrese el valor %d del arreglo: ",k+1);
        scanf("%d",&A[k]);
    }

    // Definimos apuntadores con la direccion del valor inicial y final del arreglo
    i=&A[0];
    t=&A[T-1];
    // Creamos un ciclo for para seleccionar cada valor del arreglo menos el
    ultimo, para comparar este con su valor siguiente
    for(i;i<t;i++)
    {
        // Definimos el apuntador minimo a la direccion del valor
        seleccionado (i) asumiendo que este es el menor del arreglo
        minIndex=i;
        // Creamos un ciclo for para comparar los valores siguientes (j) con
        el valor seleccionado (i)
        for(j=i+1;j<t+1;j++)
        {
            // Si el valor siguiente (j) es menor que el valor seleccionado
            (i) apuntamos el valor minimo a la direccion del valor siguiente (j)
            if(*minIndex>*j)
                minIndex=j;
        }
        // Si la direccion del valor seleccionado (i) es diferente a la del valor
        minimo, realizamos el swap
        if(i!=minIndex)
        {
            res=*minIndex;
            for(minIndex;minIndex>i;minIndex--)
            {
                *minIndex=*(minIndex-1);
            }
            *i=res;
        }
    }

    printf("\nEl arreglo ordenado es: (%d",A[0]);

```

```

        for(int k=1;k<T;k++)
            printf(" %d",A[k]);

        printf("");

        return 0;
    }

```

3. Algoritmo 3: insertar ordenar

```

#include <stdio.h>
int main()
{
    int T,*iE,*i,*f,*c,*pos,res;

    // Pedimos el tamaño del arreglo al usuario
    do
    {
        printf("Ingrese el tamaño del vector del que se sacara el valor minimo: ");
        scanf("%d",&T);
    }
    while (T<2);

    int A[T];

    // Le pedimos al usuario que llene todos los valores del arreglo
    for(int k=0;k<T;k++)
    {
        printf("\nIngrese el valor %d del arreglo: ",k+1);
        scanf("%d",&A[k]);
    }

    // Definimos 2 apuntadores a la direccion del valor inicial y final del arreglo
    i=&A[0];
    f=&A[T-1];

    // Iniciamos un ciclo for para seleccionar el elemento "Insertado" hasta que el ultimo elemento lo sea
    for(iE=&A[1];iE<=f;iE++)
    {
        // Definimos un apuntador posicion a la direccion donde el elemento seleccionado debe ir (Asumiendo que esta en el lugar correcto)
        pos=iE;
    }
}

```

```

        // Iniciamos un ciclo for para comparar el elemento "Insertado" con
        todos los valores anteriores a este en el arreglo
        for(c=iE-1;c>=i;c--)
        {
            // Si algun valor anterior al arreglo es menor al valor
            "Insertado", cambiamos el apuntador posicion para guardar la direccion en la que
            debe ir
            if(*iE<*c)
                pos=c;
        }
        // Si la direccion del apuntador posicion es diferente a la direccion
        del apuntador del valor "Insertado", hacemos el swap para acomodarlo en su lugar
        if(pos!=iE)
        {
            res=*iE;
            for(c=iE;c>pos;c--)
            {
                *c=*(c-1);
            }
            *pos=res;
        }
    }

    printf("\nEl arreglo ordenado es: (%d",A[0]);

    for(int k=1;k<T;k++)
        printf(", %d",A[k]);

    printf(")");

    return 0;
}

```