



รายงาน

เรื่อง โปรแกรม INSERTION SORT

จัดทำโดย

นายพงษ์พันธุ์ เลาวพงศ์

รหัสนักศึกษา 66543206019-2

เสนอ

อาจารย์ปิยพล ยืนยงสถาวร

รายงานนี้เป็นส่วนหนึ่งของวิชา

ENGCE124

โครงสร้างข้อมูลและขั้นตอนวิธี

(Data Structures and Algorithms)

หลักสูตร วศ.บ.วิศวกรรมคอมพิวเตอร์

สาขาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

มหาวิทยาลัยเทคโนโลยีราชมงคลล้านนา ภาคพายัพ เชียงใหม่

ภาคเรียนที่ 1 ปีการศึกษา 2567

รายงาน

เรื่อง โปรแกรม INSERTION SORT

จัดทำโดย

นายพงษ์พันธุ์ เลาวพงศ์

รหัสนักศึกษา 66543206019-2

เสนอ

อาจารย์ปิยพล ยืนยงสถาวร

รายงานนี้เป็นส่วนหนึ่งของวิชา

ENGCE124

โครงสร้างข้อมูลและขั้นตอนวิธี

(Data Structures and Algorithms)

หลักสูตร วศ.บ.วิศวกรรมคอมพิวเตอร์

สาขาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

มหาวิทยาลัยเทคโนโลยีราชมงคลล้านนา ภาคพายัพ เชียงใหม่

ภาคเรียนที่ 1 ปีการศึกษา 2567

คำนำ

รายงานฉบับนี้เป็นส่วนหนึ่งของรายวิชา ENGCE124 โครงสร้างข้อมูลและขั้นตอนวิธี (Data Structures and Algorithms) หลักสูตร วศ.บ.วิศวกรรมคอมพิวเตอร์ สาขาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์มหาวิทยาลัยเทคโนโลยีราชมงคลล้านนา ภาคพายัพ เชียงใหม่ ในระดับปริญญาตรีปีที่ 2 โดยมีจุดประสงค์ในการอธิบายโค้ดของโปรแกรม INSERTION SORT รวมถึงอธิบายหลักการทำงานของโปรแกรม INSERTION SORT และอธิบายผลลัพธ์การใช้งานโปรแกรม INSERTION SORT

ผู้จัดทำรายงานหวังว่า รายงานฉบับนี้จะเป็นประโยชน์กับผู้สนใจ หรือนักศึกษาทุกท่านที่กำลังหาศึกษาในหัวข้อของโปรแกรม INSERTION SORT หากมีข้อเสนอแนะหรือข้อผิดพลาดประการใด ผู้จัดทำขอน้อมรับไว้ และขออภัยมา ณ ที่นี้

ผู้จัดทำ

นายพงษ์พันธุ์ เลาวพงศ์

วันที่ 22/09/2567

สารบัญ

| | หน้า |
|---|------|
| คำนำ | ก |
| สารบัญ | ๗ |
| โค้ดของโปรแกรม INSERTION SORT พร้อมคำอธิบาย | 1 |
| หลักการทำงานของโปรแกรม INSERTION SORT | 5 |
| ผลลัพธ์การใช้งานโปรแกรม INSERTION SORT | 11 |
| บรรณานุกรม | 14 |

โค้ดของโปรแกรม INSERTION SORT พร้อมคำอธิบาย

```
#include <stdio.h> //ใช้ printf
#include <conio.h> //ใช้ getch
#include <stdlib.h> //ใช้ random
#include <time.h> //ใช้ time

#define MaxData 100 // กำหนดข้อมูลสูงสุด

int Data[MaxData];

int N;

void PrepareRawData(int N)
{
    int i;

    srand(time(NULL)); //เพื่อให้ได้ค่าตัวเลขสุ่มที่แตกต่างกันใน rand()

    for (i=1;i<=N;i++)

        Data[i]=1+rand() % 99; //สุ่มตัวเลขที่แตกต่างกันในช่วง 1..99
}

void DispData(int N)
{
    int i;

    for(i=1;i<=N;i++)

        printf("%2d  ",Data[i]);

    printf("\n");
}
```

โค้ดของโปรแกรม INSERTION SORT พร้อมคำอธิบาย (ต่อ)

```

void InsertionSort(int N)
{
    int i,j,temp;

    printf("-----\n");

    printf(" i ");

    for(i=1;i<=N;i++)

        printf(" (%2d)",i);

    printf("\n");

    printf("-----\n");

    printf("%2d. ",0);

    DispData(N); //แสดงข้อมูลทุกขั้นตอนการเรียงลำดับ

    for (i=1;i<N;i++) //นับ i ไปข้างหน้า
    {
        if(Data[i+1]<Data[i]) //ถ้าข้อมูลถัดไปน้อยกว่าข้อมูลก่อนหน้า
        {
            temp=Data[i+1]; //เก็บข้อมูลที่จะใส่เข้า temp

            Data[i+1]=NULL;

            j=i; //ให้ตัวนับ j วนกลับด้านหลัง

            while(temp<Data[j]) //วนลูปถ้า temp ยังคงน้อยกว่า Data[j]

            {

                Data[j+1]=Data[j]; //เลื่อนข้อมูลไปข้างหน้าไปยังบล็อกถัดไปของอาร์เรย์
            }
        }
    }
}

```

โค้ดของโปรแกรม INSERTION SORT พร้อมคำอธิบาย (ต่อ)

```

        Data[j]=NULL;

        printf("%2d. ",i+1);

        DispData(N); //แสดงทุกครั้งที่ทำการเรียงลำดับ

        j=j-1; //นับถอยหลัง j

    } //จบ while

    Data[j+1]=temp; //ใส่ temp ลงใน Data[j+1] ในที่สุด

    printf("%2d. ",i+1);

    DispData(N); //แสดงทุกครั้งที่ทำการเรียงลำดับ

} //จบ if

} //จบ for

} //จบฟังก์ชัน

int main()

{

    printf("ASCENDING INSERTION SORT\n");

    printf("=====\n");

    N=12;

    PrepareRawData(N);

    printf("Raw Data...");

    DispData(N);

    printf("Processing Data...\n");

    InsertionSort(N);

```

โค้ดของโปรแกรม INSERTION SORT พร้อมคำอธิบาย (ต่อ)

```
printf("-----\n");  
  
printf("Sorted Data : ");  
  
DispData(N); //แสดงข้อมูลที่เรียงลำดับแล้ว  
  
getch();  
  
return(0);  
  
} //จบ Main
```


หลักการทำงานของโปรแกรม INSERTION SORT

โปรแกรมการเรียงลำดับแบบแทรก (Insertion Sort) ซึ่งจะเรียงลำดับข้อมูลจากน้อยไปมาก โดยประกอบด้วยฟังก์ชันหลัก ๆ หลายฟังก์ชันที่ทำงานร่วมกันเพื่อสุมข้อมูล, แสดงข้อมูล, และเรียงลำดับข้อมูล ฟังก์ชันเหล่านี้จะอธิบายการทำงานโดยละเอียด ดังต่อไปนี้

1. การนำเข้าไลบรารี

```
#include <stdio.h> //ใช้ printf
#include <conio.h> //ใช้ getch
#include <stdlib.h> //ใช้ random
#include <time.h> //ใช้ time
```

ในส่วนของการนำเข้าไลบรารี (#include) จะมีรายละเอียดดังนี้

- <stdio.h> : ไลบรารีนี้ใช้สำหรับฟังก์ชันการรับและแสดงผลข้อมูล เช่น printf() ที่ใช้ในการพิมพ์ข้อความออกทางหน้าจอ และ scanf() ที่ใช้สำหรับการรับข้อมูลจากผู้ใช้
- <conio.h> : ไลบรารีนี้ใช้ในการทำงานกับการอินพุตจากคีย์บอร์ดในรูปแบบที่ง่ายขึ้น เช่น getch() ซึ่งใช้เพื่อรอให้ผู้ใช้งานกดปุ่มก่อนที่จะดำเนินการต่อ
- <stdlib.h> : ไลบรารีนี้มีฟังก์ชันที่เกี่ยวข้องกับการจัดการหน่วยความจำ การแปลงค่า และการสุ่ม เช่น rand() ที่ใช้สำหรับสร้างค่าตัวเลขสุ่ม
- <time.h> : ไลบรารีนี้มีฟังก์ชันที่เกี่ยวข้องกับเวลาและวันที่ เช่น time() ที่ใช้เพื่อรับค่าชั่วโมง นาที และวินาทีในรูปแบบ timestamp

2. การกำหนดค่าคงที่

```
#define MaxData 100 // กำหนดข้อมูลสูงสุด
```

ในส่วนของการกำหนดค่าคงที่ จะมีรายละเอียดดังนี้

- #define MaxData 100 : การใช้คำสั่ง #define นี้ใช้เพื่อกำหนดค่าคงที่ (constant) ในโปรแกรม โดย MaxData กำหนดค่าที่ 100 ซึ่งเป็นการกำหนดขนาดสูงสุดของอาร์เรย์ Data[] ในโปรแกรม

ค่าคงที่นี้สามารถถูกใช้ในหลายส่วนของโปรแกรม เช่น การวนลูปหรือจัดการข้อมูล เพื่อให้แน่ใจว่าอาร์เรย์ Data[] จะไม่เกินขนาดที่กำหนด (100 ข้อมูล)

3. การประกาศตัวแปร

```
int Data[MaxData];

int N;
```

ในส่วนของการประกาศตัวแปร จะมีรายละเอียดดังนี้

- int Data[MaxData] : ตัวแปร Data[] คืออาร์เรย์ชนิดจำนวนเต็ม (int) ที่สามารถเก็บข้อมูลได้มากถึง 100 ตัวเลข (เนื่องจาก MaxData = 100) ซึ่งตัวเลขในอาร์เรย์นี้จะถูกใช้ในกระบวนการสุ่ม, แสดงผล, และเรียงลำดับข้อมูลในโปรแกรม
- int N : ตัวแปร N ถูกใช้เพื่อเก็บจำนวนข้อมูลที่ต้องการให้โปรแกรมจัดการ ตัวอย่างเช่น หากเราต้องการให้โปรแกรมทำงานกับข้อมูล 12 ค่า เราจะกำหนดค่า N=12 เพื่อให้โปรแกรมทราบว่าควรสุ่มและเรียงลำดับข้อมูลกี่ค่า โดยตัวแปร N จึงเป็นตัวแปรสำคัญที่ถูกใช้ในหลายฟังก์ชันเพื่อตัดสินใจจำนวนข้อมูลที่โปรแกรมต้องทำงานด้วย

4. ฟังก์ชัน PrepareRawData

```
void PrepareRawData(int N)
{
    int i;

    srand(time(NULL)); //เพื่อให้ได้ค่าตัวเลขสุ่มที่แตกต่างกันใน rand()

    for (i=1;i<=N;i++)

        Data[i]=1+rand() % 99; //สุ่มตัวเลขที่แตกต่างกันในช่วง 1..99
}
```

ฟังก์ชัน PrepareRawData ทำหน้าที่ในการเตรียมข้อมูลดิบ (สุ่มข้อมูล) โดยขั้นตอนการทำงาน เริ่มจากรับค่า N ซึ่งเป็นจำนวนข้อมูลที่ต้องการสุ่ม จากนั้นจะใช้งานฟังก์ชัน srand(time(NULL)) เพื่อกำหนดจุดเริ่มต้นของการสุ่ม โดยใช้เวลาปัจจุบันเป็นค่าอ้างอิง ทำให้การสุ่มแต่ละครั้งมีค่าที่แตกต่างกัน ในส่วนของลูป for ทำงาน

จาก $i=1$ ไปจนถึง $i=N$ โดยในแต่ละรอบจะสุ่มค่าตัวเลขในช่วง 1 ถึง 99 โดยใช้ `rand()` แล้วเก็บค่าที่สุ่มได้ลงในอาร์เรย์ `Data[i]` และการสุ่มข้อมูลนี้จะทำให้แต่ละครั้งที่เรียกฟังก์ชัน จะได้ชุดข้อมูลที่แตกต่างกัน

5. ฟังก์ชัน DispData

```
void DispData(int N)
{
    int i;

    for(i=1;i<=N;i++)

        printf("%2d  ",Data[i]);

    printf("\n");
}
```

ฟังก์ชัน `DispData` ทำหน้าที่ในการแสดงข้อมูลที่ถูกเก็บไว้ในอาร์เรย์ `Data[]` โดยขั้นตอนการทำงานจะเริ่มจากรับค่า N เพื่อระบุจำนวนข้อมูลที่ต้องการแสดง จากนั้นจะใช้ลูป `for` เพื่อวนแสดงข้อมูลตั้งแต่ `Data[1]` ถึง `Data[N]` และข้อมูลจะแสดงในรูปแบบของตัวเลข โดยจัดรูปแบบให้เลขแต่ละตัวมีความกว้าง 2 ช่อง

6. ฟังก์ชัน InsertionSort

```
void InsertionSort(int N)
{
    int i,j,temp;

    printf("-----\n");

    printf(" i ");

    for(i=1;i<=N;i++)

        printf(" (%2d)",i);

    printf("\n");

    printf("-----\n");
```

6. ฟังก์ชัน InsertionSort (ต่อ)

```

printf("%2d. ",0);

DispData(N); //แสดงข้อมูลทุกขั้นตอนการเรียงลำดับ

for (i=1;i<N;i++) //นับ i ไปข้างหน้า
{
    if(Data[i+1]<Data[i]) //ถ้าข้อมูลถัดไปน้อยกว่าข้อมูลก่อนหน้า
    {
        temp=Data[i+1]; //เก็บข้อมูลที่จะใส่เข้า temp

        Data[i+1]=NULL;

        j=i; //ให้ตัวนับ j วนกลับด้านหลัง

        while(temp<Data[j]) //วนลูปถ้า temp ยังคงน้อยกว่า Data[j]
        {
            Data[j+1]=Data[j]; //เลื่อนข้อมูลไปข้างหน้าไปยังบล็อกถัดไปของอาร์เรย์

            Data[j]=NULL;

            printf("%2d. ",i+1);

            DispData(N); //แสดงทุกครั้งที่ทำการเรียงลำดับ

            j=j-1; //นับถอยหลัง j
        } //จบ while

        Data[j+1]=temp; //ใส่ temp ลงใน Data[j+1] ในที่สุด

        printf("%2d. ",i+1);

        DispData(N); //แสดงทุกครั้งที่ทำการเรียงลำดับ
    } //จบ if

```

6. ฟังก์ชัน InsertionSort (ต่อ)

```

    } //จบ for

} //จบฟังก์ชัน

```

ฟังก์ชัน InsertionSort ทำหน้าที่หลักในการเรียงลำดับข้อมูลจากน้อยไปมาก โดยใช้วิธีการเรียงลำดับแบบแทรก (Insertion Sort) ซึ่งเป็นการเปรียบเทียบและแทรกข้อมูลลงในตำแหน่งที่ถูกต้อง โดยหลักการทำงานเริ่มต้นด้วยการแสดงเส้นแบ่งและหัวข้อของข้อมูล จากนั้นจะพิมพ์ตำแหน่ง i ของแต่ละข้อมูล เพื่อให้สามารถเห็นการเปลี่ยนแปลงของข้อมูลในแต่ละขั้นตอน ในส่วนของลูป for จะใช้งานเพื่อวนรอบตั้งแต่ $i=1$ ไปจนถึง $i<N$ โดยในแต่ละรอบจะเปรียบเทียบค่าของข้อมูลที่ตำแหน่ง $i+1$ กับตำแหน่งก่อนหน้า ($Data[i]$) หากข้อมูลในตำแหน่ง $i+1$ น้อยกว่าข้อมูลในตำแหน่ง i จะนำค่าที่ตำแหน่ง $i+1$ เก็บไว้ในตัวแปร temp และเริ่มกระบวนการแทรกข้อมูล ในส่วนของลูป while จะใช้งานเพื่อเลื่อนข้อมูลในอาร์เรย์ไปข้างหน้า จนกว่าจะเจอตำแหน่งที่ข้อมูลใน temp ควรจะถูกแทรกเข้าไป โดยในทุกครั้งที่มีการเลื่อนข้อมูลจะพิมพ์ผลลัพธ์ออกมาเพื่อให้เห็นการเปลี่ยนแปลงของข้อมูล เมื่อเจอตำแหน่งที่เหมาะสม จะใส่ค่าที่เก็บใน temp กลับไปในอาร์เรย์

7. ฟังก์ชัน main

```

int main()

{

    printf("ASCENDING INSERTION SORT\n");

    printf("=====\n");

    N=12;

    PrepareRawData(N);

    printf("Raw Data...");

    DispData(N);

    printf("Processing Data...\n");

    InsertionSort(N);

    printf("-----\n");
}

```

7. ฟังก์ชัน main (ต่อ)

```
printf("Sorted Data : ");  
  
DispData(N); //แสดงข้อมูลที่เรียงลำดับแล้ว  
  
getch();  
  
return(0);  
  
} //จบ Main
```

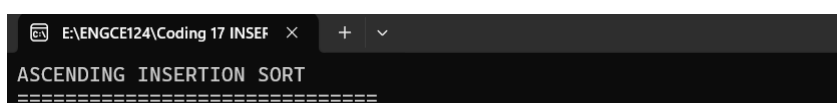
ฟังก์ชัน main เป็นจุดเริ่มต้นของโปรแกรม ซึ่งมีการทำงาน โดยเริ่มจากแสดงหัวข้อ "ASCENDING INSERTION SORT" และตั้งค่าจำนวนข้อมูล N=12 จากนั้นเรียกใช้ฟังก์ชัน PrepareRawData() เพื่อสุ่มข้อมูล 12 ค่า และแสดงข้อมูลที่สุ่มได้โดยใช้ DispData() เมื่อแสดงข้อมูลเสร็จจะแสดงข้อความ "Processing Data..." และเริ่มกระบวนการเรียงลำดับข้อมูลโดยเรียกใช้ฟังก์ชัน InsertionSort() เมื่อการเรียงลำดับเสร็จสิ้น จะแสดงข้อมูลที่เรียงแล้วโดยใช้ DispData() และรอให้ผู้ใช้กดปุ่มใด ๆ เพื่อออกจากโปรแกรม (ใช้ฟังก์ชัน getch())

ผลลัพธ์การใช้งานโปรแกรม INSERTION SORT

โปรแกรม INSERTION SORT ถูกออกแบบมาเพื่อสับข้อมูลจำนวนหนึ่งและทำการเรียงลำดับข้อมูลจากน้อยไปมาก (Ascending Order) โดยใช้วิธีการเรียงลำดับแบบแทรก (Insertion Sort)

1. เริ่มต้นโปรแกรม

เมื่อเปิดโปรแกรมขึ้นมา สิ่งแรกที่จะแสดงบนหน้าจอคือหัวข้อของโปรแกรมและเส้นแบ่งข้อความ โดยข้อความ “ASCENDING INSERTION SORT” บ่งบอกถึงชื่อของโปรแกรมและการเตรียมพร้อมสำหรับการทำงานต่อไป โดยผู้ใช้ไม่จำเป็นต้องกรอกข้อมูลใด ๆ โปรแกรมจะสับข้อมูลให้เอง



```
E:\ENGCE124\Coding 17 INSEF x + v
ASCENDING INSERTION SORT
=====
```

2. การสับข้อมูล (Raw Data)

หลังจากแสดงหัวข้อแล้ว โปรแกรมจะเริ่มสับข้อมูลจำนวน N ซึ่งในตัวอย่างนี้ถูกตั้งค่าให้ N=12 หมายถึงโปรแกรมจะสับข้อมูล 12 ค่าในช่วง 1 ถึง 99 โดยข้อมูลข้างต้นคือตัวเลขดิบที่สุ่มขึ้นมา โดยโปรแกรมใช้ฟังก์ชัน rand() ในการสุ่มค่าตัวเลขในช่วง 1 ถึง 99 ซึ่งค่าที่สุ่มได้จะเก็บไว้ในอาร์เรย์ Data[] การสุ่มนี้ทำให้แต่ละครั้งที่รันโปรแกรมจะได้ค่าที่ไม่ซ้ำกัน

```
Raw Data...90 51 51 21 2 52 89 56 42 14 83 54
```

3. การเรียงลำดับ (Insertion Sort)

หลังจากโปรแกรมสับข้อมูลแล้ว โปรแกรมจะเข้าสู่ขั้นตอนการเรียงลำดับโดยใช้วิธีการแทรก (Insertion Sort) ซึ่งจะแสดงกระบวนการเรียงลำดับทีละขั้นตอนให้เห็นอย่างชัดเจน โดยผลลัพธ์ที่ได้ สามารถอธิบายได้ดังนี้

- แถวแรก (0.): แสดงข้อมูลเริ่มต้นก่อนการเรียงลำดับ (ยังไม่ได้จัดเรียง)
- แถวต่อมา: ในแต่ละขั้นตอน โปรแกรมจะพิจารณาค่าข้อมูลที่ตำแหน่งถัดไป และเลื่อนข้อมูลไปยังตำแหน่งที่ถูกต้องตามหลักการของ Insertion Sort เช่น:
 - ในขั้นตอนที่ 2 (2.) ค่าที่ตำแหน่งที่ 3 ซึ่งคือ 17 ถูกย้ายไปยังตำแหน่งที่ถูกต้องในอาร์เรย์ ทำให้ข้อมูลเรียงลำดับใหม่จาก [5, 82, 17] เป็น [5, 17, 82]
 - ในขั้นตอนที่ 5 (5.) ค่าที่ตำแหน่งที่ 6 ซึ่งคือ 58 ถูกย้ายไปยังตำแหน่งที่เหมาะสม ส่งผลให้ข้อมูลช่วงนี้เรียงลำดับได้เป็น [5, 17, 58, 64, 73, 82]

- โปรแกรมจะทำงานต่อไปเช่นนี้จนกว่าจะเรียงข้อมูลทั้งหมดเสร็จสมบูรณ์ในขั้นตอนที่ 11 (11.) ซึ่งข้อมูลที่ได้จะเรียงจากน้อยไปมาก

Processing Data...

| i | (1) | (2) | (3) | (4) | (5) | (6) | (7) | (8) | (9) | (10) | (11) | (12) |
|-----|-------|-------|-------|-------|-------|-------|-------|-------|-------|------|------|------|
| 0. | 90 | 51 | 51 | 21 | 2 | 52 | 89 | 56 | 42 | 14 | 83 | 54 |
| 2. | 0 | 90 | 51 | 21 | 2 | 52 | 89 | 56 | 42 | 14 | 83 | 54 |
| 2. | 51 | 90 | 51 | 21 | 2 | 52 | 89 | 56 | 42 | 14 | 83 | 54 |
| 3. | 51 | 0 | 90 | 21 | 2 | 52 | 89 | 56 | 42 | 14 | 83 | 54 |
| 3. | 51 | 51 | 90 | 21 | 2 | 52 | 89 | 56 | 42 | 14 | 83 | 54 |
| 4. | 51 | 51 | 0 | 90 | 2 | 52 | 89 | 56 | 42 | 14 | 83 | 54 |
| 4. | 51 | 0 | 51 | 90 | 2 | 52 | 89 | 56 | 42 | 14 | 83 | 54 |
| 4. | 0 | 51 | 51 | 90 | 2 | 52 | 89 | 56 | 42 | 14 | 83 | 54 |
| 4. | 21 | 51 | 51 | 90 | 2 | 52 | 89 | 56 | 42 | 14 | 83 | 54 |
| 5. | 21 | 51 | 51 | 0 | 90 | 52 | 89 | 56 | 42 | 14 | 83 | 54 |
| 5. | 21 | 51 | 0 | 51 | 90 | 52 | 89 | 56 | 42 | 14 | 83 | 54 |
| 5. | 21 | 0 | 51 | 51 | 90 | 52 | 89 | 56 | 42 | 14 | 83 | 54 |
| 5. | 0 | 21 | 51 | 51 | 90 | 52 | 89 | 56 | 42 | 14 | 83 | 54 |
| 5. | 2 | 21 | 51 | 51 | 90 | 52 | 89 | 56 | 42 | 14 | 83 | 54 |
| 6. | 2 | 21 | 51 | 51 | 0 | 90 | 89 | 56 | 42 | 14 | 83 | 54 |
| 6. | 2 | 21 | 51 | 51 | 52 | 90 | 89 | 56 | 42 | 14 | 83 | 54 |
| 7. | 2 | 21 | 51 | 51 | 52 | 0 | 90 | 56 | 42 | 14 | 83 | 54 |
| 7. | 2 | 21 | 51 | 51 | 52 | 89 | 90 | 56 | 42 | 14 | 83 | 54 |
| 8. | 2 | 21 | 51 | 51 | 52 | 89 | 0 | 90 | 42 | 14 | 83 | 54 |
| 8. | 2 | 21 | 51 | 51 | 52 | 0 | 89 | 90 | 42 | 14 | 83 | 54 |
| 8. | 2 | 21 | 51 | 51 | 52 | 56 | 89 | 90 | 42 | 14 | 83 | 54 |
| 9. | 2 | 21 | 51 | 51 | 52 | 56 | 89 | 0 | 90 | 14 | 83 | 54 |
| 9. | 2 | 21 | 51 | 51 | 52 | 56 | 0 | 89 | 90 | 14 | 83 | 54 |
| 9. | 2 | 21 | 51 | 51 | 52 | 0 | 56 | 89 | 90 | 14 | 83 | 54 |
| 9. | 2 | 21 | 51 | 51 | 0 | 52 | 56 | 89 | 90 | 14 | 83 | 54 |
| 9. | 2 | 21 | 51 | 0 | 51 | 52 | 56 | 89 | 90 | 14 | 83 | 54 |
| 9. | 2 | 21 | 0 | 51 | 51 | 52 | 56 | 89 | 90 | 14 | 83 | 54 |
| 9. | 2 | 21 | 42 | 51 | 51 | 52 | 56 | 89 | 90 | 14 | 83 | 54 |
| 10. | 2 | 21 | 42 | 51 | 51 | 52 | 56 | 89 | 0 | 90 | 83 | 54 |
| 10. | 2 | 21 | 42 | 51 | 51 | 52 | 56 | 0 | 89 | 90 | 83 | 54 |
| 10. | 2 | 21 | 42 | 51 | 51 | 52 | 0 | 56 | 89 | 90 | 83 | 54 |
| 10. | 2 | 21 | 42 | 51 | 51 | 0 | 52 | 56 | 89 | 90 | 83 | 54 |
| 10. | 2 | 21 | 42 | 51 | 0 | 51 | 52 | 56 | 89 | 90 | 83 | 54 |
| 10. | 2 | 21 | 42 | 0 | 51 | 51 | 52 | 56 | 89 | 90 | 83 | 54 |
| 10. | 2 | 21 | 0 | 42 | 51 | 51 | 52 | 56 | 89 | 90 | 83 | 54 |
| 10. | 2 | 0 | 21 | 42 | 51 | 51 | 52 | 56 | 89 | 90 | 83 | 54 |
| 10. | 2 | 14 | 21 | 42 | 51 | 51 | 52 | 56 | 89 | 90 | 83 | 54 |
| 11. | 2 | 14 | 21 | 42 | 51 | 51 | 52 | 56 | 89 | 0 | 90 | 54 |
| 11. | 2 | 14 | 21 | 42 | 51 | 51 | 52 | 56 | 0 | 89 | 90 | 54 |
| 11. | 2 | 14 | 21 | 42 | 51 | 51 | 52 | 56 | 83 | 89 | 90 | 54 |
| 12. | 2 | 14 | 21 | 42 | 51 | 51 | 52 | 56 | 83 | 89 | 0 | 90 |
| 12. | 2 | 14 | 21 | 42 | 51 | 51 | 52 | 56 | 83 | 0 | 89 | 90 |
| 12. | 2 | 14 | 21 | 42 | 51 | 51 | 52 | 56 | 0 | 83 | 89 | 90 |
| 12. | 2 | 14 | 21 | 42 | 51 | 51 | 52 | 0 | 56 | 83 | 89 | 90 |
| 12. | 2 | 14 | 21 | 42 | 51 | 51 | 52 | 54 | 56 | 83 | 89 | 90 |

4. แสดงผลลัพธ์การเรียงลำดับ (Sorted Data)

เมื่อกระบวนการเรียงลำดับเสร็จสิ้น โปรแกรมจะแสดงข้อมูลที่ผ่านการเรียงลำดับแล้วพร้อมเส้นแบ่งท้ายข้อความ โดยข้อมูลข้างต้น คือผลลัพธ์สุดท้ายที่ถูกจัดเรียงจากน้อยไปมาก โดยโปรแกรมจะพิมพ์ข้อมูลเรียงตามลำดับอย่างเป็นระเบียบเรียบร้อย

```
-----
Sorted Data :  2   14   21   42   51   51   52   54   56   83   89   90
|
```


5. รอกการกดปุ่ม (End Program)

หลังจากแสดงผลลัพธ์เรียงลำดับแล้ว โปรแกรมจะหยุดทำงานชั่วคราวและรอให้ผู้ใช้กดปุ่มใด ๆ เพื่อสิ้นสุดโปรแกรม ฟังก์ชัน `getch()` จะรอกการป้อนข้อมูลจากคีย์บอร์ด ทำให้โปรแกรมไม่ปิดตัวลงทันทีหลังการแสดงผลลัพธ์สุดท้าย

```

E:\ENGCE124\Coding 17 INSEF  X  +  v

ASCENDING INSERTION SORT
=====
Raw Data...90  51  51  21  2  52  89  56  42  14  83  54
Processing Data...
-----
i  ( 1) ( 2) ( 3) ( 4) ( 5) ( 6) ( 7) ( 8) ( 9) (10) (11) (12)
-----
0. 90  51  51  21  2  52  89  56  42  14  83  54
2.  0  90  51  21  2  52  89  56  42  14  83  54
2. 51  90  51  21  2  52  89  56  42  14  83  54
3. 51  0  90  21  2  52  89  56  42  14  83  54
3. 51  51  90  21  2  52  89  56  42  14  83  54
4. 51  51  0  90  2  52  89  56  42  14  83  54
4. 51  0  51  90  2  52  89  56  42  14  83  54
4. 0  51  51  90  2  52  89  56  42  14  83  54
4. 21  51  51  90  2  52  89  56  42  14  83  54
5. 21  51  51  0  90  52  89  56  42  14  83  54
5. 21  51  0  51  90  52  89  56  42  14  83  54
5. 21  0  51  51  90  52  89  56  42  14  83  54
5. 0  21  51  51  90  52  89  56  42  14  83  54
5. 2  21  51  51  90  52  89  56  42  14  83  54
6. 2  21  51  51  0  90  89  56  42  14  83  54
6. 2  21  51  51  52  90  89  56  42  14  83  54
7. 2  21  51  51  52  0  90  56  42  14  83  54
7. 2  21  51  51  52  89  90  56  42  14  83  54
8. 2  21  51  51  52  89  0  90  42  14  83  54
8. 2  21  51  51  52  0  89  90  42  14  83  54
8. 2  21  51  51  52  56  89  90  42  14  83  54
9. 2  21  51  51  52  56  89  0  90  14  83  54
9. 2  21  51  51  52  56  0  89  90  14  83  54
9. 2  21  51  51  52  0  56  89  90  14  83  54
9. 2  21  51  51  0  52  56  89  90  14  83  54
9. 2  21  51  0  51  52  56  89  90  14  83  54
9. 2  21  0  51  51  52  56  89  90  14  83  54
9. 2  21  42  51  51  52  56  89  90  14  83  54
10. 2  21  42  51  51  52  56  89  0  90  83  54
10. 2  21  42  51  51  52  56  0  89  90  83  54
10. 2  21  42  51  51  52  0  56  89  90  83  54
10. 2  21  42  51  51  0  52  56  89  90  83  54
10. 2  21  42  51  0  51  52  56  89  90  83  54
10. 2  21  42  0  51  51  52  56  89  90  83  54
10. 2  21  0  42  51  51  52  56  89  90  83  54
10. 2  0  21  42  51  51  52  56  89  90  83  54
10. 2  14  21  42  51  51  52  56  89  90  83  54
11. 2  14  21  42  51  51  52  56  89  0  90  54
11. 2  14  21  42  51  51  52  56  0  89  90  54
11. 2  14  21  42  51  51  52  56  83  89  90  54
12. 2  14  21  42  51  51  52  56  83  89  0  90
12. 2  14  21  42  51  51  52  56  83  0  89  90
12. 2  14  21  42  51  51  52  56  0  83  89  90
12. 2  14  21  42  51  51  52  0  56  83  89  90
12. 2  14  21  42  51  51  52  54  56  83  89  90
-----
Sorted Data :  2  14  21  42  51  51  52  54  56  83  89  90

-----
Process exited after 1157 seconds with return value 0
Press any key to continue . . . |

```

บรรณานุกรม

ChatGPT. (-). Insertion Sort: Step-by-Step Process and Data Visualization. สืบค้น 22 กันยายน 2567,
จาก <https://chatgpt.com/>