



ใบงานที่ 18

เรื่อง การเรียงลำดับข้อมูล

เสนอ

อาจารย์ ปิยพล ยืนยงสถาวร

จัดทำโดย

นาย สารินทร์ อินดีะรักษา รหัส 65543206082-1

ใบงานนี้เป็นส่วนหนึ่งของรายวิชา โครงสร้างข้อมูลและขั้นตอนวิธี

หลักสูตรวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

มหาวิทยาลัยเทคโนโลยีราชมงคลล้านนา

ประจำภาคที่ 1 ปีการศึกษา 2566

คำสั่ง/คำชี้แจง

- สร้างโค้ดโปรแกรมตามตัวอย่างในเอกสารประกอบการสอนสัปดาห์ที่ 7
- แสดงโค้ดโปรแกรมเป็นส่วนๆ พร้อมอธิบาย
- แสดงผลการรันโปรแกรม พร้อมอธิบายการทำงาน
- สรุปผลการทดลอง

ลำดับขั้นตอนการทดลอง

```
int Data[MaxData];
int N;
void PrepareRawData(int N)
{
    int i;
    srand(time(NULL)); //for difference random number in rand()
    for (i=1;i<=N;i++)
        Data[i]=1+rand() % 99; //random difference number 1..99
}
void DispData(int N)
{
    int i;
    for(i=1;i<=N;i++)
        printf("%2d ",Data[i]);
    printf("\n");
}
```

- ประกาศอาเรย์ Data ที่ใช้เก็บข้อมูลตัวเลข MaxData
- int N; - ประกาศตัวแปร N
- void PrepareRawData(int N) - ฟังก์ชันนี้ใช้ในการสร้างข้อมูลที่สุ่มขึ้น โดยรับค่า N เป็นจำนวนข้อมูลที่ต้องการสร้าง
- srand(time(NULL)); - กำหนดค่าเริ่มต้นสำหรับการสุ่มตัวเลขใน rand() ด้วยค่าเวลาปัจจุบัน
- for (i = 1; i <= N; i++) - วนลูปเพื่อสร้างข้อมูลตัวเลข N ค่า และเก็บลงในอาเรย์ Data
- Data[i] = 1 + rand() % 99; - สุ่มตัวเลขในช่วง 1 ถึง 99 และเก็บลงในอาเรย์ Data ในตำแหน่ง i.
- void DispData(int N) - ฟังก์ชันนี้ใช้ในการแสดงข้อมูลที่สุ่มไป โดยรับค่า N เพื่อระบุจำนวนข้อมูลที่ต้องการแสดง
- for (i = 1; i <= N; i++) - วนลูปเพื่อแสดงข้อมูลตัวเลข N ค่า ที่อยู่ในอาเรย์ Data
- printf("%2d ", Data[i]); - แสดงค่าข้อมูลในรูปแบบตัวเลขเดียว (%2d)

```

void BubbleSort(int N)
{
    int i,j,temp;
    printf("-----\n");
    printf(" i ");
    for(i=1;i<=N;i++)
        printf(" (%2d)",i);
    printf("\n");
    printf("-----\n");
    for(i=1;i<=N-1;i++) //loop forward
    {
        if(Data[i]>Data[i+1]) //if not true position
        {
            printf("%2d. ",i+1);
            DispData(N);
            j=i+1; //loop backward
            while(Data[j]<Data[j-1]) //while if remain bubble
            {
                temp=Data[j-1]; //swap data
                Data[j-1]=Data[j];
                Data[j]=temp;
                j--; //count down j
                printf("%2d. ",i+1);
                DispData(N);
            } //end while
        } //end if
    } //end for
} //end Fn.

```

- int i, j, temp; - ประกาศตัวแปร i, j, และ temp สำหรับการนับและเก็บค่าชั่วคราว.
- for (i = 1; i <= N - 1; i++) - วนลูปเพื่อเปรียบเทียบและเรียงข้อมูลตัวเลขโดยค้นหาข้อมูลที่น้อยที่สุดที่จะไปไว้ด้านหน้าสุด.
- if (Data[i] > Data[i + 1]) - ตรวจสอบว่าข้อมูลที่อยู่ตำแหน่ง i มากกว่าข้อมูลตำแหน่ง i + 1.
- printf("%2d. ", i + 1); - แสดงข้อมูลและตำแหน่งของข้อมูลที่ย้ายตำแหน่ง.
- DispData(N); - เรียกใช้ฟังก์ชัน DispData เพื่อแสดงข้อมูลที่ปรับเปลี่ยนแล้ว.
- j = i + 1; - กำหนดค่า j เป็น i + 1 เพื่อเริ่มการเลื่อนถอยข้อมูลที่ย้ายตำแหน่งย้อนกลับ.
- while (Data[j] < Data[j - 1]) - วนลูปไปเรื่อย ๆ ถ้าข้อมูลที่ตำแหน่ง j น้อยกว่าข้อมูลที่ตำแหน่ง j - 1.
- temp = Data[j - 1]; - สร้างตัวแปรชั่วคราว temp และนำข้อมูลจากตำแหน่ง j - 1 มาเก็บ.
- Data[j - 1] = Data[j]; - ย้ายข้อมูลจากตำแหน่ง j ไปตำแหน่ง j - 1.
- Data[j] = temp; - นำข้อมูลจาก temp ที่เก็บไว้มาใส่ในตำแหน่ง j.
- j--; - ลดค่า j ลงเพื่อเริ่มการเรียงแบบ Bubble Sort ในตำแหน่งถัดไป.

```

int main()
{
printf("ASCENDING BUBBLE SORT\n");
printf("=====\\n");
N=12;
PrepareRawData(N);
printf("Raw Data : ");
DispData(N);
printf("-----\\n");
printf("Processing Data...\\n");
BubbleSort(N);
printf("-----\\n");
printf("Sorted Data : ");
DispData(N); //Sorted Data
getch();
return(0);
} //End Main

```

- printf("ASCENDING BUBBLE SORT\\n"); - แสดงข้อความ "ASCENDING BUBBLE SORT" บนหน้าจอ.
- N = 12; - กำหนดค่า N เป็น 12 โดยระบุจำนวนข้อมูลที่ต้องการสร้างและเรียง.
- PrepareRawData(N); - เรียกใช้ฟังก์ชัน PrepareRawData เพื่อสุ่มและเตรียมข้อมูลที่จะเรียง.
- printf("Raw Data : "); - แสดงข้อความ "Raw Data : " เพื่อบอกว่าจะแสดงข้อมูลก่อนการเรียง.
- DispData(N); - เรียกใช้ฟังก์ชัน DispData เพื่อแสดงข้อมูลตัวเลขที่สุ่มได้.
- BubbleSort(N); - เรียกใช้ฟังก์ชัน BubbleSort เพื่อเรียงลำดับข้อมูลในอาร์เรย์ Data.
- printf("Sorted Data : "); - แสดงข้อความ "Sorted Data : " เพื่อบอกว่าจะแสดงข้อมูลหลังการเรียง.
- DispData(N); - เรียกใช้ฟังก์ชัน DispData เพื่อแสดงข้อมูลหลังการเรียงลำดับ.
- getch(); - รอผู้ใช้กดปุ่มใดๆ (ใช้ getch()) เพื่อปิดหน้าต่างของโปรแกรม.
- return(0); - จบการทำงานของฟังก์ชัน main และโปรแกรม.

ผลลัพธ์การทดลอง

```
C:\Users\Sarin\Desktop\ENG0 < X + v
ASCENDING BUBBLE SORT
=====
Raw Data : 31 91 14 27 31 84 67 11 4 21 48 89
-----
Processing Data...
-----
i  ( 1) ( 2) ( 3) ( 4) ( 5) ( 6) ( 7) ( 8) ( 9) (10) (11) (12)
-----
3. 31 91 14 27 31 84 67 11 4 21 48 89
3. 31 14 91 27 31 84 67 11 4 21 48 89
3. 14 31 91 27 31 84 67 11 4 21 48 89
4. 14 31 91 27 31 84 67 11 4 21 48 89
4. 14 31 27 91 31 84 67 11 4 21 48 89
4. 14 27 31 91 31 84 67 11 4 21 48 89
5. 14 27 31 91 31 84 67 11 4 21 48 89
5. 14 27 31 31 91 84 67 11 4 21 48 89
6. 14 27 31 31 91 84 67 11 4 21 48 89
6. 14 27 31 31 84 91 67 11 4 21 48 89
7. 14 27 31 31 84 91 67 11 4 21 48 89
7. 14 27 31 31 84 67 91 11 4 21 48 89
7. 14 27 31 31 67 84 91 11 4 21 48 89
8. 14 27 31 31 67 84 91 11 4 21 48 89
8. 14 27 31 31 67 84 11 91
```

สรุปผลการทดลอง

โปรแกรมนี้ใช้ในการสับและเรียงลำดับข้อมูลตัวเลขแบบน้อยไปหามากโดยใช้อัลกอริทึม Bubble Sort และแสดงผลลัพธ์แต่ละขั้นตอนของการเรียงลำดับ.

สื่อ / เอกสารอ้างอิง

ไฟล์ประกอบการสอนของ อาจารย์ ปิยพล ยืนยงสถาวร เรื่อง : การเรียงลำดับข้อมูล