Report
Assignment 4 : Linked List

จัดทำโดย

นายสรธร แก้วโชติช่วงกูล 63070501067 CPE REGULAR

รายงานนี้เป็นส่วนหนึ่งของวิชา CPE111

Programming With Data Structures

King Mongkut's University of Technology Thonburi

- **สิ่งที่ทำใน Assignment**

1. สร้างฟังก์ชั่นต่างๆที่ทำงานเกี่ยวกับ linked list เช่น ฟังก์ชั่นในการสร้างโหนดของ linked list, ฟังก์ชั่นในการ add ข้อมูล linkedlist แบบต่างๆ, ฟังก์ชั่นในการเรียงข้อมูลใน linkedlist, ฟังก์ชั่นในการดูข้อมูลที่ตำแหน่ง(index)ที่ต้องการ, ฟังก์ชั่นในการลบข้อมูลใน linkedlist, ฟังก์ชั่นในการดูข้อมูลที่ต้องการว่าอยู่ตำแหน่งใด และฟังก์ชั่นการคำนวณทางคณิตศาสตร์ต่างๆ
2. สร้างฟังก์ชั่นในการทำงานเกี่ยวกับ String เพื่อรับค่าของคำสั่งเข้ามาทำงานเช่น ฟังก์ชั่นในการ spilt string จาก white space เป็นต้น
3. สร้างฟังก์ชั่นในการตรวจสอบค่า syntax หรือ group ในการทำงานของ string ที่รับมาและต้องเช็คเงื่อนไขต่างๆในการทำงานของแต่ละ syntax ด้วย
4. ฝึกการคิดเกี่ยวกับการทำงาน pointer ต่างๆต้องคอพอัพเดต Pointer ตลอดเวลาและต้องคิดว่าต้องลบหรือเพิ่มข้อมูล Pointer จะชี้ไปที่ใดต่อไป

- Source code

- ฟังก์ชั่นที่ทำงานเกี่ยวกับ Linked list

➢ Create node (สร้างโหนดใหม่)

```
linkedlist *create_node(double value){ // ฟังก์ชั่นในการสร้าง Node แต่ละอันของ linkedlist
    linkedlist *ptr; // จอง pointer มาตัวหนึ่ง
    ptr = (linkedlist *) malloc(sizeof(linkedlist)); // จองพื้นที่ของ Memory ที่จะเก็บ node นั้นๆ
    ptr->info = value;      // ใส่ค่าของข้อมูลลงในโหนด
    ptr->next = NULL; // ปรับให้มันชี้ไปที่ NULL
    return ptr; // ส่ง address กลับ
}
```

➢ Add data (เพิ่มข้อมูลเข้าไปในตัวสุดท้ายของ linkedlist)

```
void add_node(double value){ // ฟังก์ชั่นเพิ่มข้อมูล linkedlist ที่ตัวสุดท้าย
    linkedlist *ptr; // จอง Pointer ขึ้นมาตัวหนึ่ง
    ptr = create_node(value); // ให้มันไปสร้างโหนด จอง memory มารอไว้
    if(first == NULL){ // เช็คว่าข้อมูลใน Linkedlist มีหรือป่าว
        first = last = ptr; // ไม่มีให้ first last ชี้ไปที่ข้อมูลตัวที่จะเพิ่มเข้ามา
    }
    else{
        last->next = ptr; // แต่ถ้ามีอยู่แล้วให้เพิ่มไปตัวสุดท้ายของ linkedlist
        last = ptr;
    }
}
```

➢ Insert data(เพิ่มข้อมูลเข้าไปโดยหาตำแหน่งที่ข้อมูลควรอยู่โดยเรียงจากน้อยไปมาก)

```
void insert_node(double value,int pos){ // เพิ่มข้อมูลแบบเรียงลำดับให้ด้วย ต้องการ ตำแหน่งที่จะให้เพิ่มและค่าข้อมูล
    int i = 0;
    linkedlist *prev; // pointer เอาไว้ชี้ตัวก่อนหน้าที่จะ insert ข้อมูลเข้าไป
    prev = first; // กำหนดในมันชี้ไปที่ตัวแรกก่อนที่จะวนรอบเพื่อหาตำแหน่งของมัน
    if(pos == 0) // ถ้ามันเท่ากับตำแหน่งแรก
        push_node(value); // ใช้ Push ช่วย
    else if(pos == (size_of_linkedlist())) //สุดท้าย
        add_node(value); // ใช้ add ช่วย
    else{
        while(i != pos-1){ // อยู่ตรงกลาง Linkedlist หมายเหตุ สำคัญต้องเช็คเงื่อนไขและต้องดูว่า i เริ่มจากเท่าไหร่ถ้า i เริ่มจาก 0 != pos-1 แต่ถ้า i=1 จะได้ i != pos
            prev = prev->next; // วนรอบหาตำแหน่งตัวก่อนหน้าที่จะเพิ่มข้อมูลเข้าไป
            i++; // นับ i
        }
        ptrglo = create_node(value); // สร้างโหนดรอ
        ptrglo->next = prev->next; // เปลี่ยนทิศทาง Pointer ที่ชี้หลังจากเจอตำแหน่ง previous
        prev->next = ptrglo;
    }
}
```

➢ Push (การเพิ่มข้อมูลเข้าไปใน Linkedlist ในตำแหน่งหน้าสุด)

```c
void push_node(double value){ // สร้างโหนดและผลักมันเข้าไปใน Linkedlist ในตำแหน่งแรก
    linkedlist *ptr;
    ptr = create_node(value); // สร้างโหนด
    if(first == NULL) // เช็คว่ามีข้อมูลไหม
        first = last = ptr; // ไม่มี first last ชี้ไปที่ตัวนั้น
    else{
        ptr->next = first; // ถ้ามี
        first = ptr; // ก็เพิ่มไปตัวแรกให้มันชี้ไปที่ first ชี้และเปลี่ยน First เป็นตัวเอง
    }
}
```

➢ Peek (การดูข้อมูลในตำแหน่งนั้นๆ

```c
void peek_node(int index){ // ดูข้อมูลที่ Index นั้นๆ
    int count = 0,i=0,size;
    size = size_of_linkedlist()-1;
    if(index == 0){  //ดูข้อมูลที่ตำแหน่งแรก
        ptrglo = first;
        printf("Ans : %g\n",ptrglo->info);
    }
    else if(index == -1 || (index == size_of_linkedlist()-1)){ // ดูข้อมูลตำแหย่งสุดท้าย
        ptrglo = first;
        for(i=0;i<size;i++){
            ptrglo = ptrglo->next;
        }
        printf("Ans : %g\n",ptrglo->info);
    }
    else{ // ดูข้อมูลที่อยู่ระหว่างกลาง
        ptrglo = first;
        while(i<index && ptrglo->next != NULL){ // ต้องวนรอบไปเรื่อยๆจนกว่าจะถึง Index ที่ต้องการดูข้อมูล
            ptrglo = ptrglo->next;
            i++;
        }
        if(ptrglo->next != NULL) // เช็คว่ามันวนไปเรื่อยๆแล้วเจอไหมในกรณีนี้คือเจอ
            printf("Ans : %g\n",ptrglo->info);
        else // อันนี้คือไม่เจอ แปลว่ามันค้นไปเลยตัวสุดท้ายแล้วก็ยังไม่ถึง index ที่อยาดูค่า
            printf("Don't have index to peek,have only [0]-[%d]\n",size); // index เกิน
    }
}
```

➢ Delete (ลบข้อมูลในตำแหน่งที่ต้องการ)

```
128  void delete_position(int pos){ // Node ของ Linkedlist ตรง Index ที่ต้องการลบ
129      linkedlist *ptr,*prev;
130      int i=0;
131      double value;
132      ptr = prev = first;
133      if(pos == 0){ //ลบตัวแรก
134          first = ptr->next;
135          ptr->next = NULL;
136      }
137      else if(pos == 0 && size_of_linkedlist() == 1) // มีข้อมูลตัวเดียว
138          first = last = NULL;
139      else{
140          while(i != pos-1){
141              prev = prev->next;
142              i++;
143          }
144          if(pos == (size_of_linkedlist()-1)){ // ลบตัวสุดท้าย
145              ptr = last;
146              last = prev;
147              prev->next = NULL;
148          }
149          else{ // ลบตัวกลาง
150              ptr = prev->next;
151              prev->next = ptr->next;
152              ptr->next = NULL;
153          }
154      }
155      free(ptr); // ลบหน่วยความจำที่ไม่ได้ใช้แล้วคืนให้ระบบ
156  }
```

➢ Search (หาตำแหน่งของข้อมูลที่ทราบค่า)

```
113  int search_index(double value,int index[]){ // หา Index ของข้อมูลที่เราทราบค่าอยู่แล้ว
114      int i=0,count=0;
115      ptrglo = first;
116      while(ptrglo != NULL){ // วนรอบไปเรื่อยๆจนกว่าจะหมด
117          if(ptrglo->info == value){ // ถ้าเจอค่าที่ต้องการดู Index ให้เก็บค่า Index ที่เจอลงใน Array เพราะอาจมีมากกว่าตำแหน่งเดียว
118              index[count++] = i; // พร้อมกับนับจำนวนด้วยจะได้ทราบค่าเจอกี่ที่
119          }
120          ptrglo = ptrglo->next;
121          i++; // ตัวบอก Index ที่เจอว่ามันเท่ากับเท่าไหร่
122      }
123      if(count==0) // ถ้ามัน = 0 คือไม่เจอเลย
124          return -1; // บอกว่าไม่เจอ
125      else // ถ้าไม่เท่ากับ 0 แสดงว่าเจอ
126          return count; // ให้บอกว่าเจอกี่ตัว
127  }
```

➢ Sort (เรียงลำดับข้อมูลจากน้อย -> มาก)

```
209 void scansort_linkedlist(){ // เรียงลำดับข้อมูลใน Linkedlist ใช้ pointer
210     linkedlist *ptr_i,*ptr_j;
211     double swap;
212     for(ptr_i=first; ptr_i!=NULL ; ptr_i = ptr_i->next){
213         for(ptr_j=ptr_i->next; ptr_j != NULL ; ptr_j = ptr_j->next){
214             if(ptr_i->info > ptr_j->info){
215                 swap = ptr_i->info;                 // เรียงเหมือน scansort ปกติทุกอย่างต่างที่ว่าใช้ Address ในการวนรอบไปเรื่อยๆ
216                 ptr_i->info = ptr_j->info;          // แล้วดูข้อมูลใน node แล้วเอามาเปรียบเทียบเอา
217                 ptr_j->info = swap;
218 
219             }
220         }
221     }
222 }
```

➢ Pop (ดึงข้อมูลตัวหน้าสุดของ Linkedlist ออกมา)

```
103 linkedlist *pop_node(double *value){ // ดึงหรือลบข้อมูลตัวแรกสุดใน linkedlist
104     linkedlist *ptr;
105     ptr = first;
106     *value = ptr->info;
107     first = first->next;
108     ptr->next = NULL;
109     if(first == NULL) //มีข้อมูลตัวเดียวแล้วกำลังจะถูกดึงออก
110         last = NULL;  // แสดงว่าเมื่อดึงออกแล้วจะไม่มีข้อมูลหรือใน Linkedlist
111     return ptr; // ส่ง address กลับเพื่อนำไปใช้งานต่อ
112 }
```

➢ Help (แสดงคำสั่งที่สามารถทำงานได้)

```
234 void show_command(){ // ฟังก์ชันแสดงคำสั่งที่สามารถใช้งานได้
235     printf("\t\t\tCommand : (don't need anything behind command): [end][list][help][sort][pop][sqrt][rec][neg][pow][+][-][*][/]\n");
236     printf("\t\t\tCommand : (need only one number behind command): [peek n][search n][push n][delete n] \n");
237     printf("\t\t\tCommand : (need at least one number behind command): [insert n][add n] \n");
238     printf("\t\tNote! : n instead of number\n");
239 }
```

- ฟังก์ชั่นที่ทำงานเกี่ยวกับ String

  - Split String (แยกสตริงออกมาเป็นตัวๆและนับจำนวน)

```
int spilt(char *buff,char token[][20]) //ฟังก์ชั่นแยกแต่ละตัวเพื่อจะนำไปบอกว่าแต่ละตัวนั้นเป็นสตริงชนิดอะไร
{
    char *tok; // สร้าง pointer เพื่อชี้ตัวที่ต้องการแยก
    int count = 0; // สร้างตัวแปร จน.เต็มเพื่อเก็บค่าจำนวน
    tok = strtok(buff," "); // เป็นการให้ tok ชี้ไปที่ตำแหน่งเริ่มและเพิ่มไปเรื่อยๆโดยหา token ของสตริงจนกว่าจะเจอเว้นวรรคจึงหยุดโดยจะเติม \0 ให้อัตโนมัติ
    while(tok != NULL){ // ทำงานเมื่อ tok ไม่เท่ากับ \0
        strcpy(token[count++],tok); // copy ค่าสตริงที่อ่านมาได้มาเก็บไว้ใน token เพื่อจะเอาไปแสดงผลต่อ
        tok = strtok(NULL," "); //เช็คต่อจาก null ที่มันทำให้ก่อนหน้าเพราะ buff เพี้ยนไปแล้ว
    }
    return count; // รีเทิร์นค่าของ count กลับไปใน main เพิ่มนำไปวนรอบเช็คชนิดของสตริงต่อ
}
```

- ฟังชั่นทั่วไปใน Assignment

  - Check group (แยกประเภทของคำสั่ง)

```
int check_group(char *token){ // ฟังก์ชั่นเช็คว่าเป็นฟังก์ชั่นไหน
    char fname[19][10] = {"end","list","sort","pop","help","sqrt","rec","neg","pow","+","-","*","/","delete","search","peek","push","add","insert"}; // สร้าง array 2 มิติมาเก็บค่าสตริงที่เป็นฟังก์ชั่น
    int i=0,check =-1; // กำหนด ตัวแปรจำนวนเต็ม
    char buff[20]; //  สร้างสตริงเพื่อจะเอามาเปรียบเทียบ
    strcpy(buff,token); // copy จากสตริงที่รับมาเปรียบไปอีกสตริงหนึ่ง
    strlwr(buff); // ให้ทุกตัวในสตริงเป็นตัวพิมเล็ก
    for(i=0;i<19;i++){ // สำหรับรอบเพื่อเช็คกลุ่มของสตริงตัวแรกที่รับมา
        if(strcmp(fname[i],buff) == 0)
            check = i; // ถ้าตรงกับตัวไหนให้รีเทิร์นค่าตำแหน่งตัวนั้นมา
    }
    if(check < 0) // ไม่ตรงสักอัน
        return 0;
    else if (check>=0 && check <=12) // อยู่ในกลุ่มที่ 1
        return 1;
    else if (check>=13 && check <=16) // อยู่ในกลุ่ม 2
        return 2;
    else return 3; // กลุ่ม 3 ตั้งแต่ตัวที่ 17 ไป
}
```

  - Do command list 1 (ฟังก์ชั่นในการทำงานของคำสั่งประเภทที่ 1)

```
void Do_command_list1(char token[][20],int count){ // ฟังก์ชั่นในการทำงานของ Syntax กลุ่ม 1
    double value=0,value2=0;
    linkedlist *ptr;
    if(strcmp(token[0],"list") == 0){ // ถ้าสตริงเท่ากับ list
        if(size_of_linkedlist() > 0) // เช็คว่ามีข้อมูลใน Linkedlist หรือไม่
            print_nodeall(); // มีแสดงข้อมูลทั้งหมด
        else // ไม่มีบอกว่าไม่มีข้อมูล
            printf("No data now!.\n");
        }
    else if(strcmp(token[0],"sort") == 0){    // ถ้าสตริงเท่ากับ sort
        if(size_of_linkedlist() > 0){ // เช็คว่ามีข้อมูลใน Linkedlist หรือไม่
            scansort_linkedlist(); // มี
            printf("Ans : Sort success!.\n");
        }
        else
            printf("No data now!.\n"); // ไม่มีบอกไม่มีข้อมูล
        }
    else if(strcmp(token[0],"pop") == 0){     // ถ้าสตริงเท่ากับ pop
        if(size_of_linkedlist() > 0){// เช็คว่ามีข้อมูลใน Linkedlist หรือไม่
            ptrglo = pop_node(&value); // มีก็ให้ pop ออกมา
            printf("Ans : %g\n",value); // แสดงค่าที่ Pop
            free(ptrglo); // ลบ memmory ทิ้ง
        }
        else
            printf("No data now!.\n"); // ไม่มีบอกไม่มีข้อมูล
        }
    else if(strcmp(token[0],"help") == 0){    // ถ้าสตริงเท่ากับ help
        show_command(); // แสดงผลคำสั่งที่ใช้งานได้
    }
```

```
299         else if(strcmp(token[0],"sqrt") == 0){  // ถ้าสตริงเท่ากับ sqrt
300             if(size_of_linkedlist() > 0){ // เช็คว่ามีข้อมูลใน Linkedlist หรือไม่
301                 ptr = pop_node(&value); //  มีก็ Pop ข้อมูลตัวแรกออกมา
302                 ptr->info = sqrt(value); // เอามาคำนวณ
303                 printf("Ans : %g\n",ptr->info); //แสดงคำตอบ
304                 push_node(ptr->info); // ใส่กลับ Linkedlist
305                 free(ptr); // คืนความจำให้ระบบ
306             }
307             else
308                 printf("No data now!.\n"); // ไม่มีบอกไม่มีข้อมูล
309         }
310         else if(strcmp(token[0],"rec") == 0){  // ถ้าสตริงเท่ากับ rec
311             if(size_of_linkedlist() > 0){ // เช็คว่ามีข้อมูลใน Linkedlist หรือไม่
312                 ptr = pop_node(&value); //  มีก็ Pop ข้อมูลตัวแรกออกมา
313                 ptr->info = 1/value; // เอามาคำนวณ
314                 printf("Ans : %g\n",ptr->info); //แสดงคำตอบ
315                 push_node(ptr->info); // ใส่กลับ Linkedlist
316                 free(ptr); // คืนความจำให้ระบบ
317             }
318             else
319                 printf("No data now!.\n"); // ไม่มีบอกไม่มีข้อมูล
320             }
```

```
321     else if(strcmp(token[0],"neg") == 0){ // ถ้าสตริงเท่ากับ neg
322         if(size_of_linkedlist() > 0){ // เช็คว่ามีข้อมูลใน Linkedlist หรือไม่
323             ptr = pop_node(&value); //  มีก็ Pop ข้อมูลตัวแรกออกมา
324             ptr->info = -value; // เอามาคำนวณ
325             printf("Ans : %g\n",ptr->info); //แสดงคำตอบ
326             push_node(ptr->info); // ใส่กลับ Linkedlist
327             free(ptr); // คืนความจำให้ระบบ
328         }
329         else
330             printf("No data now!.\n"); // ไม่มีบอกไม่มีข้อมูล
331         }
332     else if(strcmp(token[0],"pow") == 0){   // ถ้าสตริงเท่ากับ pow
333         if(size_of_linkedlist() > 1) { // เช็คว่ามีข้อมูลใน Linkedlist มากกว่า 1 ตัวหรือไม่
334         ptr = pop_node(&value);
335         ptrglo = pop_node(&value2);   //  มีก็ Pop ข้อมูลออกมา 2 ครั้งเพราะต้องการ 2 ตัวเพื่อคำนวณ
336         ptr->info = pow(value2,value); // เอามาคำนวณ
337         printf("Ans : %g\n",ptr->info);  //แสดงคำตอบ
338         push_node(ptr->info);  // ใส่กลับ Linkedlist
339         free(ptr);free(ptrglo); // คืนความจำให้ระบบ
340         }
341         else if(size_of_linkedlist() == 1) // ถ้ามีแค่ 1 ตัวให้บอกต้องการข้อมูล 2 ตัว
342             printf("Need 2 data to Calculate\n");
343         else
344             printf("No data now!.\n"); // ไม่มีบอกไม่มีข้อมูล
345         }
346     else if(strcmp(token[0],"+") == 0){  // ถ้าสตริงเท่ากับ +
347         if(size_of_linkedlist() > 1){ // เช็คว่ามีข้อมูลใน Linkedlist มากกว่า 1 ตัวหรือไม่
348         ptr = pop_node(&value);
349         ptrglo = pop_node(&value2); //  มีก็ Pop ข้อมูลออกมา 2 ครั้งเพราะต้องการ 2 ตัวเพื่อคำนวณ
350         ptr->info = value2 + value;  // เอามาคำนวณ
351         printf("Ans : %g\n",ptr->info); //แสดงคำตอบ
352         push_node(ptr->info);  // ใส่กลับ Linkedlist
353         free(ptr);free(ptrglo);   // คืนความจำให้ระบบ
354         }
355         else if(size_of_linkedlist() == 1) // ถ้ามีแค่ 1 ตัวให้บอกต้องการข้อมูล 2 ตัว
356             printf("Need 2 data to Calculate\n");
357         else
358             printf("No data now!.\n"); // ไม่มีบอกไม่มีข้อมูล
359
360         }
```

```
361         else if(strcmp(token[0],"-") == 0){ // ถ้าสตริงเท่ากับ -
362             if(size_of_linkedlist() > 1){ // เช็คว่ามีข้อมูลใน Linkedlist มากกว่า 1 ตัวหรือไม่
363             ptr = pop_node(&value);
364             ptrglo = pop_node(&value2); //  มีก็ Pop ข้อมูลออกมา 2 ครั้งเพราะต้องการ 2 ตัวเพื่อคำนวณ
365             ptr->info = value2 - value; // เอามาคำนวณ
366             printf("Ans : %g\n",ptr->info);  //แสดงคำตอบ
367             push_node(ptr->info);  // ใส่กลับ Linkedlist
368             free(ptr);free(ptrglo); // คืนความจำให้ระบบ
369             }
370             else if(size_of_linkedlist() == 1) // ถ้ามีแค่ 1 ตัวให้บอกต้องการข้อมูล 2 ตัว
371                 printf("Need 2 data to Calculate\n");
372             else
373                 printf("No data now!.\n"); // ไม่มีบอกไม่มีข้อมูล
374             }
375         else if(strcmp(token[0],"*") == 0){ // ถ้าสตริงเท่ากับ *
376             if(size_of_linkedlist() > 1){ // เช็คว่ามีข้อมูลใน Linkedlist มากกว่า 1 ตัวหรือไม่
377             ptr = pop_node(&value);
378             ptrglo = pop_node(&value2); //  มีก็ Pop ข้อมูลออกมา 2 ครั้งเพราะต้องการ 2 ตัวเพื่อคำนวณ
379             ptr->info = value2*value; // เอามาคำนวณ
380             printf("Ans : %g\n",ptr->info); //แสดงคำตอบ
381             push_node(ptr->info); // ใส่กลับ Linkedlist
382             free(ptr);free(ptrglo); // คืนความจำให้ระบบ
383             }
384             else if(size_of_linkedlist() == 1) // ถ้ามีแค่ 1 ตัวให้บอกต้องการข้อมูล 2 ตัว
385                 printf("Need 2 data to Calculate\n");
386             else
387                 printf("No data now!.\n"); // ไม่มีบอกไม่มีข้อมูล
388             }
389         else{ // ถ้าไม่ใช่เหลืออันเดียว /
390             if(size_of_linkedlist() > 1){ // เช็คว่ามีข้อมูลใน Linkedlist มากกว่า 1 ตัวหรือไม่
391             ptr = pop_node(&value);
392             ptrglo = pop_node(&value2); //  มีก็ Pop ข้อมูลออกมา 2 ครั้งเพราะต้องการ 2 ตัวเพื่อคำนวณ
393             ptr->info = value2/value; // เอามาคำนวณ
394             printf("Ans : %g\n",ptr->info); //แสดงคำตอบ
395             push_node(ptr->info); // ใส่กลับ Linkedlist
396             free(ptr);free(ptrglo); // คืนความจำให้ระบบ
397             }
398             else if(size_of_linkedlist() == 1) // ถ้ามีแค่ 1 ตัวให้บอกต้องการข้อมูล 2 ตัว
399                 printf("Need 2 data to Calculate\n");
400             else
401                 printf("No data now!.\n"); // ไม่มีบอกไม่มีข้อมูล
402             }
403     }
```

➢ Do command list 2 (ฟังก์ชั่นในการทำงานของคำสั่งประเภทที่ 2)

```
404  void Do_command_list2(char token[][20],int count){
405      int i=0,start=0,index[20],indexnum=0,j=0;
406      double value;
407      char ch;
408      if(strcmp(token[0],"peek") == 0){ // ถ้าเป็นคำสั่ง Peek
409          if(size_of_linkedlist() > 0){ // เช็คข้อมูลใน linkedlist ว่ามีไหม
410              indexnum = (int)string_to_num(token[1]); // มีก็แปลงค่าเลขที่ตามมา 1 ตัว
411              peek_node(indexnum); // ส่งค่านั้นไป peek
412          }
413          else printf("No data now!.\n"); // ไม่มีข้อมูล
414      }
415      else if(strcmp(token[0],"search") == 0){ // ถ้าเป็นคำสั่ง Search
416          if(size_of_linkedlist() > 0){ // เช็คข้อมูลใน linkedlist ว่ามีไหม
417              value = string_to_num(token[1]);  // มีก็แปลงค่าเลขที่ตามมา 1 ตัว
418              indexnum = search_index(value,index); // ส่งค่านั้นไป หา Index
419              if(indexnum == -1) // ไม่มีค่านั้นใน Linkedlist
420                  printf("Not Found %g in Linkedlist.\n",value);
421              else{ // มี
422                  printf("Found %g in Linkedlist : ",value);
423                  for(i=0;i<indexnum;i++){ // วนรอบบอกว่าเจอที่ไหนบ้างเพราะอาจมีมากกว่า 1 ตำแหน่ง
424                      if(i < indexnum-1)
425                          printf("%d",index[i]);
426                      else
427                          printf("%d\n",index[i]);
428                  }
429              }
430          }
431          else printf("No data now!.\n"); // ไม่มีข้อมูลใน Linkedlist
432      }
433      else if(strcmp(token[0],"push") == 0){ // คำสั่ง Push
434          value = string_to_num(token[1]); // แปลงค่าสตริงเป็นเลข
435          push_node(value); // push
436      }
```

```
437      else{ //delete
438          value = string_to_num(token[1]);// แปลงค่าสตริงเป็นเลข
439          indexnum = search_index(value,index); //  เช็คค่านั้นว่ามีใน linkedlist ไหม
440          if(indexnum != -1){ // มี
441              while(indexnum != 0){ // วนรอบทำไปเรื่อยๆจนกว่าจะเท่ากับค่าว่าเจอกี่ตัว
442                  printf("Ans : Do you want to delete linklist [%d] (Y/N)  ",index[0]); // ถามว่าจะลบค่านั้นจริงๆไหม
443                  scanf(" %c",&ch);  // รับคำตอบ
444                  if(ch == 'y' || ch == 'Y'){ //ลบ
445                      delete_position(index[0]); // ส่งค่าไปลบ
446                      if(indexnum > 1) // เช็คว่าถ้าจำนวนครั้งที่จะต้องทำมากกว่า 1
447                          monitor_delete(); // ให้อัพเดตข้อมูลใน linkedlist ด้วย
448                      indexnum -= 1; // ลบจำนวนครั้งที่ต้องทำไป 1 เพราะทำไปแล้ว
449                      for(i=1,j=0;i<=indexnum;i++) // อัพเดตข้อมูล index ที่ต้องลบ
450                          index[j++] = index[i]-1; // อัพเดตตำแหน่ง index เพราะมันถูกลบ
451                  }
452                  else{ // ไม่ลบ
453                      printf("Ans : Cancel delete %g in Linkedlist[%d]\n",value,index[0]); // ไม่แสดงข้อความยกเลิก
454                      indexnum -= 1; // ลบจำนวนครั้งที่ต้องทำไป 1 เพราะทำไปแล้ว
455                      for(i=1,j=0;i<=indexnum;i++) // อัพเดตข้อมูล index ที่ต้องลบ
456                          index[j++] = index[i];
457                  }
458                  rewind(stdin); // ลบค่าที่ค้างในคีย์บอร์ด
459              }
460          }
461          else if(indexnum == -1 && (size_of_linkedlist() == 0)) // ถ้าไม่เจอค่าที่ต้องการลบแล้วไม่มีข้อมูลใน Linkedlist
462              printf("Ans : No data to delete now.\n",value); // บอกไม่มีข้อมูล
463          else // แต่ถ้ามีข้อมูลแต่หาตำแหน่งไม่เจอก็คือไม่มีข้อมูลนั้นใน likedlist
464              printf("Ans : Not Found %g in Linkedlist.\n",value);
465      }
466  }
```

➢ Do command list 3 (ฟังก์ชั่นในการทำงานของคำสั่งประเภทที่ 3)

```
467  void Do_command_list3(char token[][20],int count){ // คำสั่ง group 3
468      int i,check,pos=0;
469      double value=0;
470      if(strcmp(token[0],"add") == 0){ // คำสั่ง add
471          for(i=1;i<count;i++){ // วนรอบ add ข้อมูลไปเรื่อยจนกว่าพารามิเตอร์ที่ตามหลังมาจะหมด
472              value = string_to_num(token[i]); // แปลงค่าก่อน
473              add_node(value);
474          }
475      }
476      else{ // คำสั่ง insert
477          check = checksort_linkedlist(); // check ว่าข้อมูลเรียงอยู่หรือไม่
478          if(check == 1){ //เรียง
479              for(i=1;i<count;i++){
480                  value = string_to_num(token[i]);  // แปลงค่าก่อน
481                  pos = Index_to_insert(value); // หาตำแหน่งที่ควรจะ Add ข้อมูลลงไป
482                  insert_node(value,pos); // ส่งตำแหน่งไปเพิ่ม
483              }
484          }
485          else // ไม่เรียง
486              printf("Can't Insert data,please sorted before.\n");
487
488      }
489  }
```

Note! อธิบายโค้ด comment อยู่ในตัวโค้ด

- Test case & อธิบาย

```
List : NULL
Command : pip
Syntax Error!,please try again.
List : NULL
Command : delete
Parametor Error!
List : NULL
Command : add
Parametor Error!
List : NULL
Command : pop 10
Parametor Error!
List : NULL
Command : add 10.1.1
Parametor Error!
List : NULL
Command : add 3x
Parametor Error!
List : NULL
Command : delete 1
Ans : No data to delete now.
List : NULL
Command : neg
No data now!.
List : NULL
Command : +
No data now!.
List : NULL
Command : add 30 20 10 20
List :  30 20 10 20
Command : peek 0
Ans : 30
List :  30 20 10 20
Command : peek 1
Ans : 20
List :  30 20 10 20
Command : peek 4
Don't have index to peek,have only [0]-[3]
List :  30 20 10 20
Command : peek -1
Ans : 20
List :  30 20 10 20
Command : search 30
Found 30 in Linkedlist : 0
```

```
List :  30 20 10 20
Command : search 50
Not Found 50 in Linkedlist.
List :  30 20 10 20
Command : delete 30
Ans : Do you want to delete linklist [0] (Y/N)  y
List :  20 10 20
Command : delete 100
Ans : Not Found 100 in Linkedlist.
List :  20 10 20
Command : delete 20
Ans : Do you want to delete linklist [0] (Y/N)  n
Ans : Cancel delete 20 in Linkedlist[0]
Ans : Do you want to delete linklist [2] (Y/N)  y
List :  20 10
Command : delete 10
Ans : Do you want to delete linklist [1] (Y/N)  y
List :  20
Command : delete 20
Ans : Do you want to delete linklist [0] (Y/N)  y
List : NULL
Command : push 10
List :  10
Command : push 20
List :  20 10
Command : pop
Ans : 20
List :  10
Command : pop
Ans : 10
List : NULL
Command : pop
No data now!.
List : NULL
Command : sort
No data now!.
List : NULL
Command : rec
No data now!.
List : NULL
Command : sqrt
No data now!.
List : NULL
Command : add 9.5 50.5 20 -5 -20 2
List :  9.5 50.5 20 -5 -20 2
Command : +
Ans : 60
List :  60 20 -5 -20 2
```

```
Command : -
Ans : -40
List :  -40 -5 -20 2
Command : *
Ans : 200
List :  200 -20 2
Command : /
Ans : -0.1
List :  -0.1 2
Command : rec
Ans : -10
List :  -10 2
Command : neg
Ans : 10
List :  10 2
Command : pow
Ans : 1024
List :  1024
Command : sqrt
Ans : 32
List :  32
Command : +
Need 2 data to Calculate
List :  32
Command : -
Need 2 data to Calculate
List :  32
Command : *
Need 2 data to Calculate
List :  32
Command : /
Need 2 data to Calculate
List :  32
Command : pow
Need 2 data to Calculate
List :  32
Command : pop
Ans : 32
List : NULL
Command : insert 300
List :  300
Command : insert 100
List :  100 300
Command : insert 800 400
List :  100 300 400 800
Command : add 200
List :  100 300 400 800 200
```

```
Command : add 200
List :  100 300 400 800 200
Command : insert 500
Can't Insert data,please sorted before.
List :  100 300 400 800 200
Command : sort
Ans : Sort success!.
List :  100 200 300 400 800
Command : insert 500
List :  100 200 300 400 500 800
Command : help
                    Command : (don't need anything behind command): [end][list][help][sort][pop][sqrt][rec][neg][pow][+][-][*][/]
                    Command : (need only one number behind command): [peek n][search n][push n][delete n]
                    Command : (need at least one number behind command): [insert n][add n]
               Note! : n instead of number
List :  100 200 300 400 500 800
Command : end
Answer : END
This program is written by Sorathorn Kaewchotchuangkul 63070501067 CPE REGULAR
```

- **สรุปความเข้าใจของตนเอง**

ใน Assignment นี้จะเป็นการทำงานเกี่ยวกับ pointer ในภาษาซีซึ่งต้องใช้ความเข้าใจในระดับนึงของเกี่ยวกับ Pointer เพื่อที่จะสามารถทำ Assignment นี้ให้สำเร็จได้เพราะต้องคิดอยู่ตลอดเวลาสำหรับการทำงานที่มี Pointer เข้ามาเกี่ยวข้องต้องคิดว่าถ้าเกิดทำสิ่งนี้ไปแล้วจะเกิดอะไรขึ้นและ Pointer ควรเป็นยังไงต่อไป เป็นต้น นอกจากนี้จะต้องคิดถึงขอบเขตในการทำงานต่างๆด้วยว่าเราจะให้มันทำงานกี่รอบวนรอบไปจนเจออะไรถึงหยุดทำ ในจุดนี้เป็นจุดที่สำคัญอีกจุดหนึ่งเพราะถ้าคิดผิดพลาดแล้วอาจจะทำให้ผลลัพธ์ที่ออกมานั้นเกิดการผิดพลาดได้

- **ผลการประเมินตนเอง**

ให้ตนเองอยู่ที่ระดับ 60 เพราะใช้เวลาค่อนข้างนานมากในการทำโจทย์นี้เนื่องจากยังไม่เข้าใจในเรื่อง pointer อย่างถูกต้องและยังมีบางเรื่องที่เข้าใจผิดเกี่ยวกับ pointer ทำให้งานนั้นเกิดการผิดพลาดและต้องแก้ไขทำให้ใช้เวลาในการทำ Assignment นี้นานตามข้างต้นที่ได้กล่าวไปและส่งไปตรงตามกำหนดที่อาจารย์ได้ระบุไว้ใน LEB 2 ด้วยแต่หลังจากทำโจทย์แล้วก็ทำให้เข้าใจในเรื่อง Pointer มากขึ้นกว่าเดิมในจุดที่เข้าใจผิดก็ทราบว่าทำไมถึงไม่เป็นแบบนั้นและมีความมั่นใจในการทำโจทย์แนวนี้มากขึ้นกว่าเดิม