Rajshahi University of Engineering & Technology
Department of Computer Science of Engineering

EXPERIMENT NO: 05
NAME OF EXPERIMENT: Linked Lists

SUBMITTED TO:

RIZOAN TOUFIQ
ASSISTANT PROFESSOR
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
RAJSHAHI UNIVERSITY OF ENGINEERING & TECHNOLOGY

SUBMITTED BY:

NAME: MD. ARIFUL ISLAM
ROLL NO.: 1803046
GROUP: 2ND THIRTY
DATE OF EXP.: 30-09-2019
DATE OF SUB. : 14-09-2019
SERIES: 18 SERIES

MACHINE CONFIGURATION:

ASUS X510UF
CORE I5 8TH GEN PROCESSOR
UP TO 3.4 GHZ
8 GB RAM
OS WIN 10

INDEX

THEORY: A **header linked list** is a linked list which always contains a special node, called the at the beginning of the list. The following are two kinds of widely used header lists:

1. A **grounded header list** is a header list where the last node contains the null pointer.

2. A **circular header list** is a header list where the last node points back to the header node.

A **two-way list** is a linear collection of data elements,called nodes,where each nodes N is divided into **three** parts:

1. An information field **INFO** which contains the data of N.

2. A pointer field **FORW** which contains the location of the next node in the list.

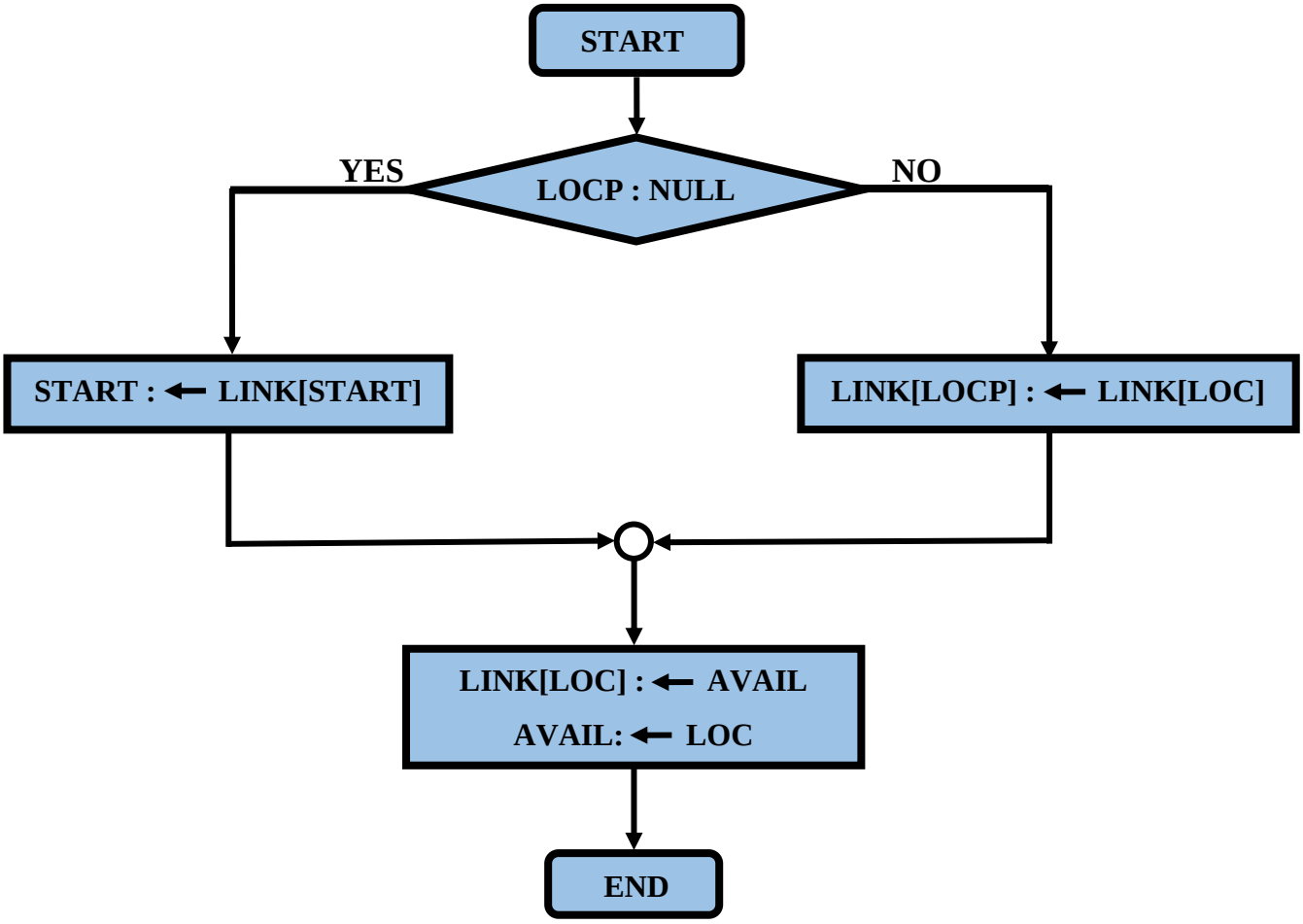
3. A pointer field **BACK** which contains the location of the preceding node in the list.

Operations on Two-Way Lists:

1. **Traversing**
2. **Searching**
3. **Deleting**
4. **Inserting etc.**

PROBLEM 1: Deleting a given node from a linked list.

FLOW CHART:



ALGORITHM: DEL (INFO, LINK, START, AVAIL, LOC, LOCP)

This algorithm delete the node N with location LOC. LOCP is the location of the node which precedes N or, when N is the first node, LOCP = NULL.

1. IF LOCP=NULL then:
   Set START:=LINK[START]. [Delete First Node]
   Else:
   Set LINK[LOCP]:=LINK[LOC] [Delete N node]
   [End of IF structure]
2. LINK [LOC] = AVAIL and AVAIL := LOC
3. Exit.

CODE:

```
#include<stdio.h>

int main(){
    char info[12]={'\0','\0','U','E','C','R','T','E','\0','\0','S','\0'};
    int link[12]={10,'\0',8,'\0',11,3,5,7,1,12,4,2};
    int loc=4,locp=11,start=6,avail=1,ptr;
    ptr=start;
    while(ptr!='\0'){
        printf("%d %c %d\n",ptr,info[ptr-1],link[ptr-1]);
        ptr=link[ptr-1];
    }
    if(locp=='\0')
        start=link[start-1];
    else
        link[locp-1]=link[loc-1];
    link[loc-1]=avail;
    avail=loc;
    ptr=start;
    while(ptr!='\0'){
        printf("\n%d %c %d",ptr,info[ptr-1],link[ptr-1]);
        ptr=link[ptr-1];
    }
    return 0;
}
```

OUTPUT:

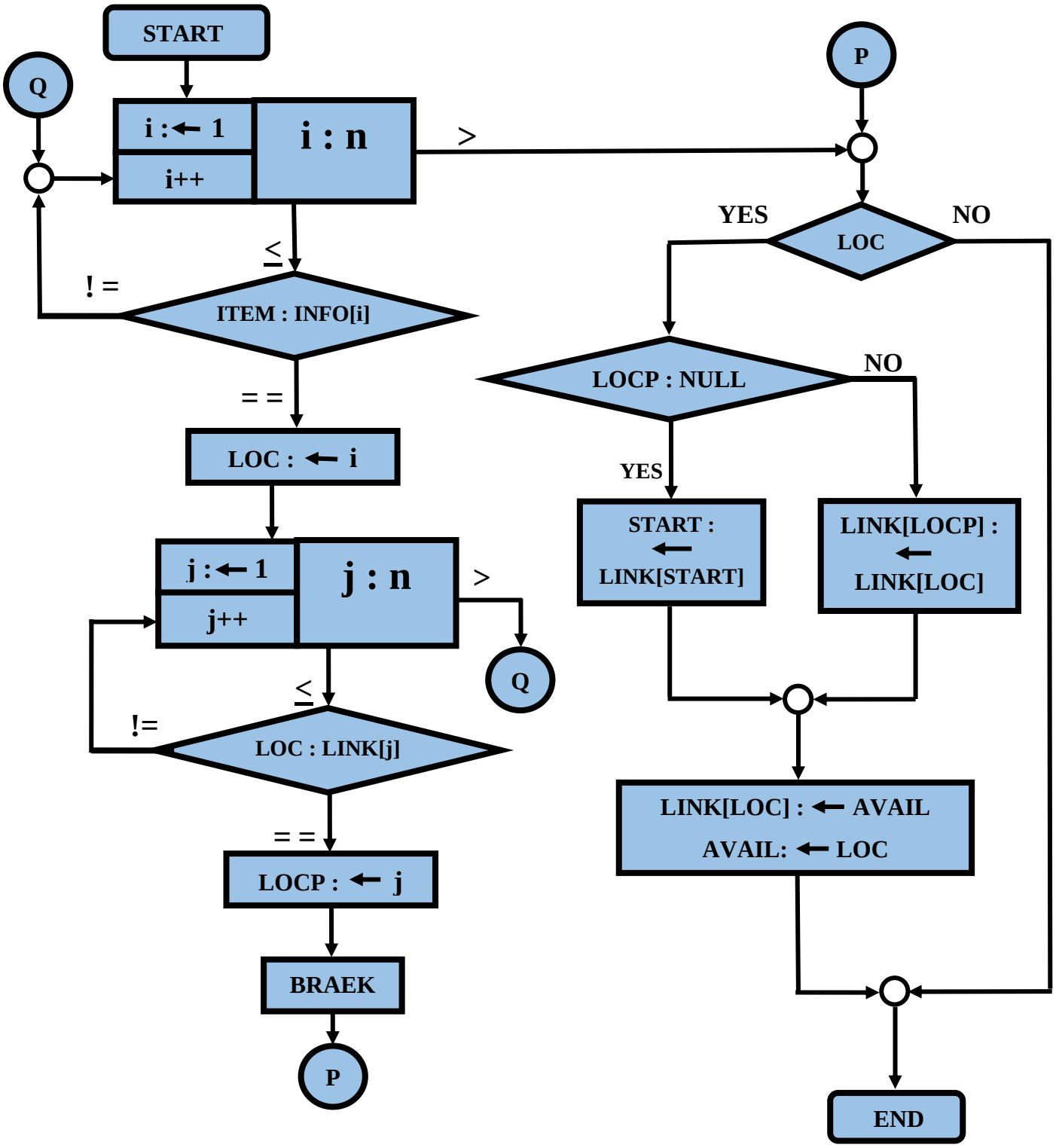
```
"F:\2nd Semester\CSE\CSE.1202\Lab 5\1 Delete Node (Given Node).exe"
6 R 3
3 U 8
8 E 7
7 T 5
5 C 11
11 S 4
4 E 0

6 R 3
3 U 8
8 E 7
7 T 5
5 C 11
11 S 0

Process returned 0 (0x0)   execution time : 0.145 s
Press any key to continue.
```

PROBLEM 2: Deleting a node with a given ITEM of information.

FLOW CHART:



ALGORITHM: :

DEL (INFO, LINK, START, ITEM,AVAIL, LOC, LOCP)

This algorithm delete the node N which contain information ITEM. At first it finds the location LOC of ITEM. It also finds the location LOCP of the node which precedes N or, when N is the first node, LOCP = NULL.

1. Repeat for i = 1 to n by 1
2. If ITEM=INFO[i] then:
   Set LOC=i.
   [End of IF structure]
3. Repeat for j=1 to n by 1
4. If LOC=LINK[j] then:
   Set LOCP=j and Break.
   [End of IF structure]
   [End of Step 3 loop]
   Break.
   [End of Step 1 loop]
5. If LOC !=True then: Exit.
   Else :
6. If LOCP=NULL then:
   Set START:=LINK[START]. [Delete 1st Node containing ITEM]
   Else:
   Set LINK[LOCP]:=LINK[LOC] [Delete N node containing ITEM]
   [End of IF structure]
7. LINK [LOC] = AVAIL and AVAIL := LOC
   [End of Step 5 IF structure]
8. Exit.

CODE:

```
#include<stdio.h>

int main(){
    char info[12]={'\0','\0','U','E','C','R','T','E','\0','\0','S','\0'},item='T';
    int link[12]={10,'\0',8,'\0',11,3,5,7,1,12,4,2};
    int i,j,loc,locp,start=6,avail=1,ptr;
    ptr=start;
    while(ptr!='\0'){
        printf("%d %c %d\n",ptr,info[ptr-1],link[ptr-1]);
        ptr=link[ptr-1];
      }
    for(i=0;i<12;i++){
        if(item==info[i]){
            loc=i+1;
            for(j=0;j<12;j++){
                if(loc==link[j]){
                    locp=j+1;
                    break;
                  }
                }
              break;
            }
      }

    if(locp=='\0')
        start=link[start-1];
    else
        link[locp-1]=link[loc-1];

    link[loc-1]=avail;
    avail=loc;
    ptr=start;
    while(ptr!='\0'){
        printf("\n%d %c %d",ptr,info[ptr-1],link[ptr-1]);
        ptr=link[ptr-1];
     }

    return 0;
}
```

Output :

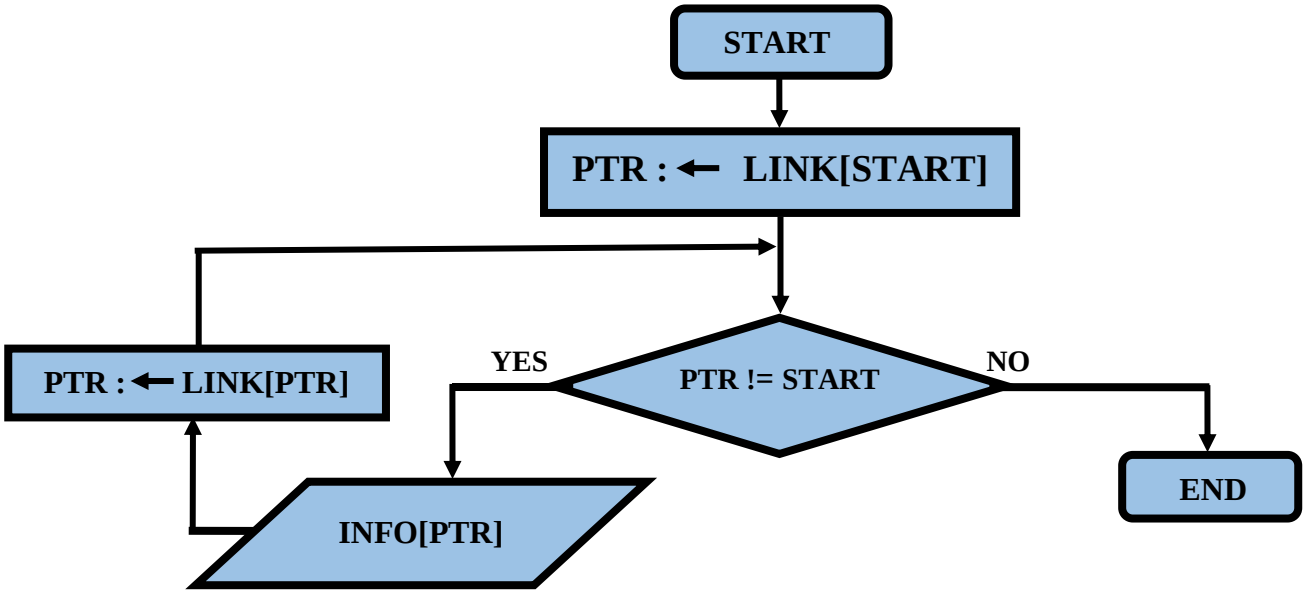
```
"F:\2nd Semester\CSE\CSE.1202\Lab 5\2 Delete Node (Given Info).exe"
6 R 3
3 U 8
8 E 7
7 T 5
5 C 11
11 S 4
4 E 0

6 R 3
3 U 8
8 E 5
5 C 11
11 S 4
4 E 0

Process returned 0 (0x0)   execution time : 0.112 s
Press any key to continue.
```

PROBLEM 3: Traversing a Circular Header List.

FLOW CHART:



ALGORITHM: Let LIST be a circular header list in memory. This algorithm traverses LIST, applying an operation PROCESS to each node of LIST.

1. Set PTR:=LINK[START].
2. Repeat Steps 3 and 4 while PTR ≠ START
3. Apply PROCESS to INFO[PTR].
4. Set PTR:=LINK[PTR].
   [End of step 2 loop]
5. Exit.

CODE:

```
#include<stdio.h>

int main()
{
   char info[12]={'\0','\0','U','E','C','R','T','E','\0','\0','S','\0'};
   int link[12]={10,'\0',8,6,11,3,5,7,1,12,4,2};
   int start=6,avail=1,ptr;

   printf("Current\tInfo\tNext\n");
   ptr=link[start-1];
   while(ptr!=start)
   {
      printf("%d \t %c \t %d\n",ptr,info[ptr-1],link[ptr-1]);
      ptr=link[ptr-1];
   }

   return 0;
}
```

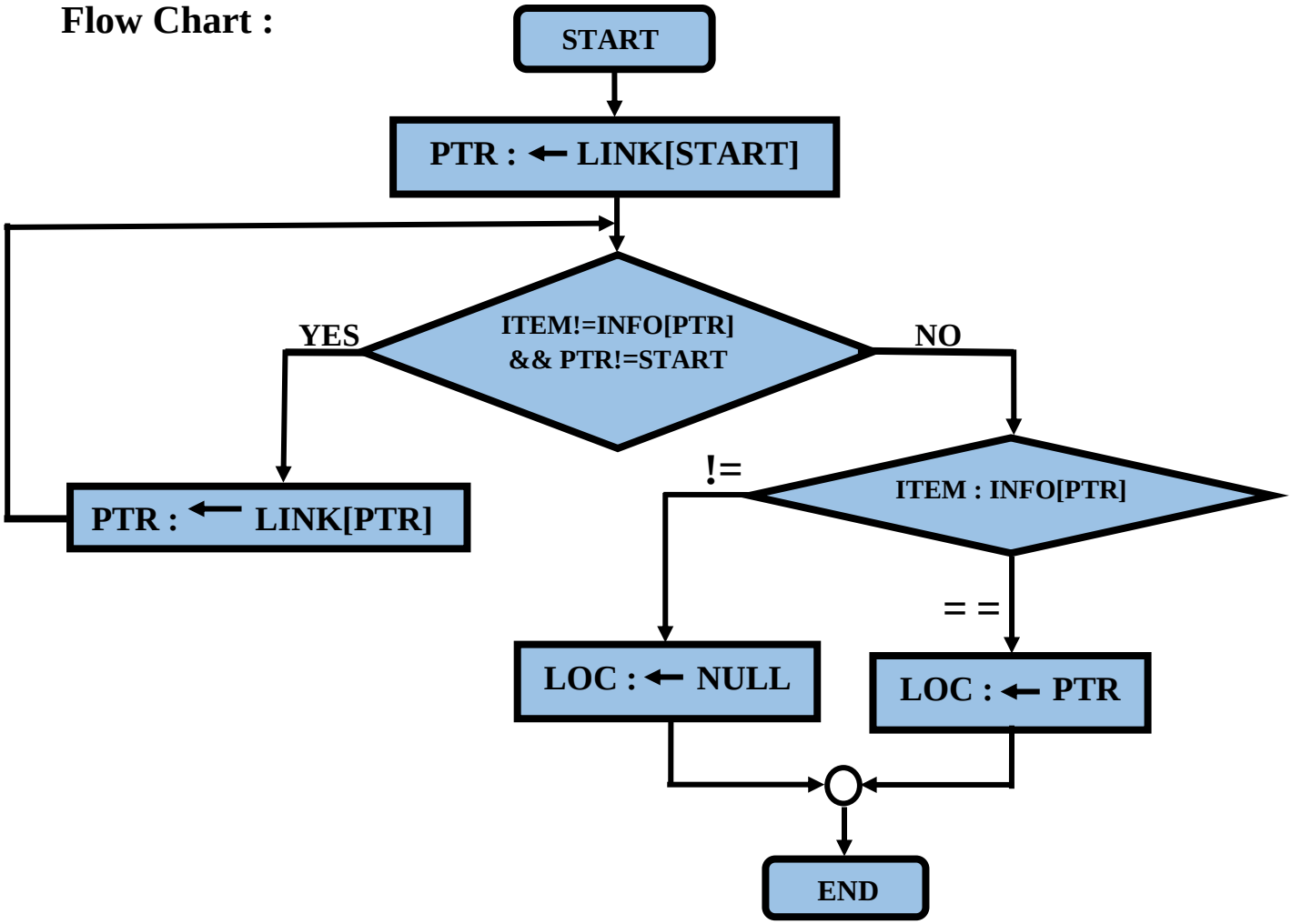
OUTPUT :

```
"F:\2nd Semester\CSE\CSE.1202\Lab 5\3 Traversing a Circular HL.exe"
Current Info    Next
3        U       8
8        E       7
7        T       5
5        C       11
11       S       4
4        E       6

Process returned 0 (0x0)   execution time : 0.096 s
Press any key to continue.
```

PROBLEM 4: Finding the location of the first node in circular header list when contains ITEM.

Flow Chart :



ALGORITHM: SEARCH (INFO, LINK, START, ITEM, LOC)

LIST is a linked list in memory. This algorithm finds the location LOC of the node where ITEM first appears in LIST, or sets LOC-NULL.

1. Set PTR:= LINK[START]
2. Repeat while INFO[PTR]≠ITEM & PTR≠START

   Set PTR =LINK[PTR]

   [End of Repeat 2 loop]
3. If ITEM = INFO[PTR] then:

   Set LOC:=PTR

   Else:

   SET PTR := NULL [Search is unsuccessful].

   [End of If statement]
4. Exit.

CODE:

```
#include<stdio.h>

int main(){
    char info[12]={'\0','\0','U','E','C','R','T','E','\0','\0','S','\0'},item='E';
    int link[12]={10,'\0',8,6,11,3,5,7,1,12,4,2};
    int start=6,avail=1,ptr,loc;
    ptr=link[start-1];
    while(ptr!=start){
        printf("%d \t %c \t %d\n",ptr,info[ptr-1],link[ptr-1]);
        ptr=link[ptr-1];
    }
    ptr=link[start-1];
    while(info[ptr-1]!=item&&ptr!=start){
        ptr=link[ptr-1];
    }
    if(info[ptr-1]==item)
         loc=ptr;
    else
        loc='\0';
    printf("\nLocation: %d\n",loc);

    return 0;
}
```

OUTPUT:

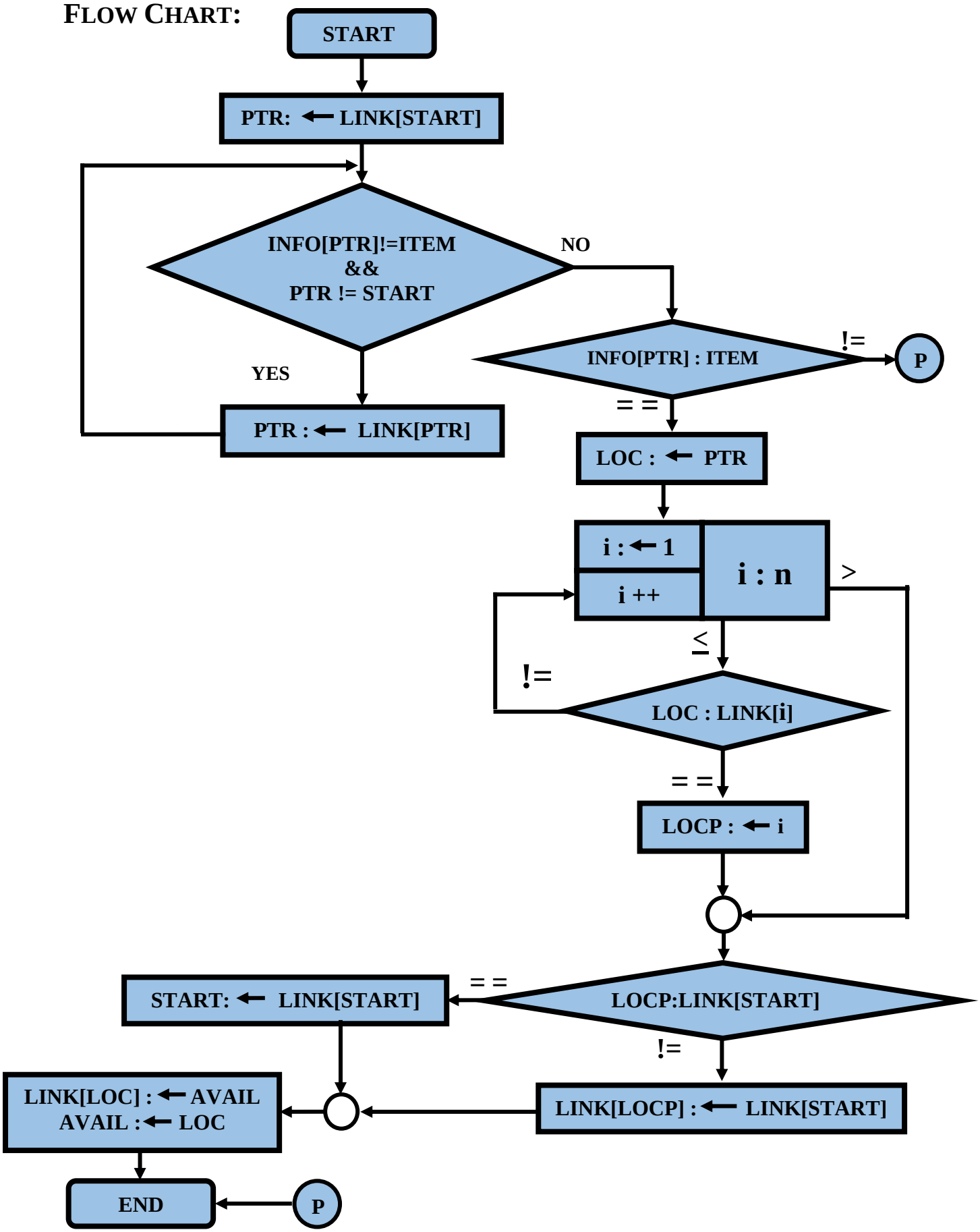
```
"F:\2nd Semester\CSE\CSE.1202\Lab 5\4 LOC of ITEM.exe"
3       U       8
8       E       7
7       T       5
5       C       11
11      S       4
4       E       6

Location: 8

Process returned 0 (0x0)   execution time : 0.155 s
Press any key to continue.
```

PROBLEM 5: Deleting the node in circular header list with a given ITEM of information.

FLOW CHART:



ALGORITHM:

DEL (INFO, LINK, START, ITEM,AVAIL, LOC, LOCP)

This algorithm delete the node N which contain information ITEM. At first it finds the location LOC of ITEM. It also finds the location LOCP of the node which precedes N or, when N is the first node, LOCP = LINK[START].

1. Set PTR:= LINK[START]
2. Repeat while INFO[PTR]≠ITEM & PTR≠START

   Set PTR =LINK[PTR]

   [End of Repeat 2 loop]
2. If ITEM ≠ INFO[PTR] then:

   Exit.

   Else:

   Set LOC:=PTR

   [End of IF structure]
4. Repeat for i=1 to n by 1
5. If LOC=LINK[i] then:

   Set LOCP=i and Break.

   [End of IF structure]

   [End of Step 4 loop]
6. If LOCP=LINK[START] then:

   Set START:=LINK[START]. [Delete 1st Node containing ITEM]

   Else:

   Set LINK[LOCP]:=LINK[LOC] [Delete N node containing ITEM]

   [End of IF structure]
7. LINK [LOC] = AVAIL and AVAIL := LOC.
8. Exit.

CODE:

```
#include<stdio.h>

int main(){
    char info[12]={'\0','\0','U','E','C','R','T','E','\0','\0','S','\0'},item='U';
    int link[12]={10,'\0',8,6,11,3,5,7,1,12,4,2};
    int i,j,loc,start=6,locp=link[start-1],avail=1,ptr;

    ptr=link[start-1];
    while(info[ptr-1]!=item&&ptr!=start){
        ptr=link[ptr-1];}
    if(info[ptr-1]==item){
        loc=ptr;
        for(i=0;i<12;i++){
            if(loc==link[i]){
                locp=i+1;
                break;
            }}}
    if(locp==link[start-1])
        start=link[start-1];
    else
        link[locp-1]=link[loc-1];
    link[loc-1]=avail;
    avail=loc;

    return 0;
}
```

OUTPUT:

```
"F:\2nd Semester\CSE\CSE.1202\Lab 5\5 Delete Node of CHL (Given Info).exe"
3        U        8
8        E        7
7        T        5
5        C        11
11       S        4
4        E        6

8        E        7
7        T        5
5        C        11
11       S        4
4        E        6

Process returned 0 (0x0)   execution time : 0.093 s
Press any key to continue.
```

DISCUSSION: The problems were based on Linked List. After trying sometime I had solved the first five problems. But still **I have some confussion** in the problems of **two way linked list** and for that I could not complete these problems.

The End