JUNAID. GIRKAR
60004190057
JAGirkar

END SEM-3 EXAM
DATA STRUCTURES

09.12.2020

Q3

```
#include <stdio.h>
# include <malloc.h>
# include <conio.h>

struct node
{
        int data ;
        int priority;
        struct node *next ;
};
struct node *start = NULL ;
struct node *insert (struct node *);
struct node *delete (struct node *);
void display (struct node *);

int main ()
{
        int option ;
        clrscr () ;
        do
        {
            printf ("\n MAIN MENU");
            printf ("\n 1. Insert \n 2. Delete \n 3. Display \n 4. Exit
            \n Enter your option : ");
            scanf ("%d", &option );
        switch (option)
        {
            case 1:
                start = insert (start);
                break;
            case 2:
                start = delete (start);
                break;
            case 3 :
                display (start);
                break;
        }
    } while (option != 4);
}

struct node *insert (struct node *start)
{
    int val, pri ;
    struct node *ptr, *p;
    ptr = (struct node *) malloc (sizeof (struct node));
    printf ("Enter the value and its priority : ");
    scanf (" %d %d", &val ,&pri);
    ptr -> data = val ;
    ptr -> priority = pri ;

    if (start == NULL || pri < start -> priority)
    {
        ptr -> next = start ;
        start = ptr ;
    }
```

```
            else
            {
                p = start ;
                while ( p -> next != NULL && p -> next -> priority <= pri)
                        p = p -> next
                ptr -> next = p -> next ;
                p -> next = ptr ;
            }
            return start ;
    }
    struct node * delete (struct node * start)
    {
            struct node * ptr ;
            if (start == NULL)
            {
                printf ("\n UNDERFLOW");
                return ;
            }
            else
            {
                ptr = start ;
                printf ("\n Deleted Item : %d", ptr -> data) ;
                start = start -> next ;
                free (ptr) ;
            }
            return start ;
    }
```

```
void display(struct node * start)
{
        struct node * ptr ;
        ptr = start ;
        if (start == NULL)
                printf("\n QUEUE IS EMPTY");
        else
        {
            printf ("\n PRIORITY QUEUE IS : ");
            while (ptr != NULL)
            {
                printf("\t %d [priority = %d]", ptr -> data,
                ptr -> priority);
                ptr = ptr -> next ;
            }
        }
}
```

JUNAID. GIRKAR
60004190057
JAGirkar

Q4

```
#include <stdio.h>

int  partition ( int  a[], int  beg, int end) {
     int  piv  =  a [end];
     int  i  = beg , j ;
     for ( j = beg ; j<end ; j++)
     {
          if (a[j] < piv) {
               int  temp  =  a [j] ;
               a[j]  =  a [i] ;
               a[i]  =  temp;
               i++ ;
          }
     }
     int  temp  =  a [end] ;
     a [end]  =  a [i] ;
     a [i]  =  temp ;
     return  i;
}
void  quickSort (int a[], int  l ,int h)  {
     if (l <h){
          int  i = partition (a, l, h) ;
          quick Sort  (a, l, i-1) ;
          quickSort (a , i+1 , h) ;
     }
}
```

```
int  main ()
{
     int  array [6] = {30,20,10,40,50}, n = 5, c ;
     quicksort (array, 0, n-1) ;
     printf ("Sorted list") ;
     for (c=0; c<n ; c++)
          printf (" %d", array [c]) ;
     return 0;
}
```

Given array : 25, 10, 7, 30, 15, 2, 96, 14

Step 1 :   beg = 0
end = n-1 = 7
pivot = 14
jindex = 0
-i = 0
∴ a[i] = 25   a[i] > pivot
∴ NO change
-i = 1
∴ a[i] = 10   a[i] < pivot
∴ a[i] swapped with a[pindex]
↓ 10   ↓ 25

∴ Array : 10, 25, 7, 30, 15, 2, 96, 14

3.1

JUNAID. GIRKAR
60004190057
JAGirkar

- i = 2
a[i] = 7 a[i] < pivot
∴ a[i] swapped with a[pindex] & pindex = 2
↓ ↓
7 25

∴ Array: 10, 7, 25, 30, 15, 2, 96, 14.

- i = 3
∴ a[i] = 30 a[i] > pivot
∴ No change.
- i = 4
∴ a[i] = 15 ∴ a[i] > pivot
∴ No change
- i = 5
a[i] = 2 a[i] < pivot
∴ a[i] swapped with a[pindex] & pindex = 3
↓ ↓
2 25

Array: 10, 7, 2, 30, 15, 25, 96, 14

∴ - i = 6
a[i] = 96 a[i] > pivot
Now swapped a[pindex] with a[end];
Array: 10, 7, 2, 14, 15, 25, 96, 30
pindex = 3

STEP 2: beg = 0
end = 2
pivot = 2
pindex = 0

JUNAID · GIRKAR
60004190057
JAGirkar

- -i = 0

  a [i] = 10   a[i] > pivot ∴ No change

- -i = 1

  a [i] = 7   a [i] > pivot ∴ No change

Now, swap a [pindex] with a [end]
∴ Array = 2, 7, 10, 14, 15, 25, 96, 30

beg = 4
end = 7
pivot = 30
pindex = 4

- -i = 4

  a[i] = 15   a [i] < pivot

  ∴ swap ~~a~~ a[i] with a[pindex]

  ∴ pindex = 5

∴ Array : 2, 7, 10, 14, 15, 25, 96, 30

- -i = 5

  a[i] = 25   a [i] < pivot

  ∴ swap a[i] & a [pindex]

  pindex = 6

Array : 2, 7, 10.14, 15, 25, 96, 30

3.2

JUNAID. GIRKAR
60004190057
JAGirkar

- i = 6

a[i] = 96   a[i] > pivot   No change

Now swap a[pindex] with a[end]

∴ Array : 2, 7, 10, 14, 15, 25, 30, 96

STEP 3 :

beg = 0
end = -1
∵ end < beg
∴ Terminates

beg = 7
end = 7
∵ end = beg
∴ Terminates

SORTED ARRAY : 2, 7, 10, 15, 25, 30, 96

JUNAID · GIRKAR
60004190057
JAGirkar

Q7

```
# include <stdio.h>
# include <stdlib.h>

struct node
{
        char data ;
        struct node * left ;
        struct node * right ;
};
struct node * root = NULL ;
struct node *stk [100] ;
int top = -1 ;

void push (struct node * temp)
{
        if (top == 99)
        {
            printf("stack is full");
        }
        else
        {
            top ++ ;
            stk [top] = temp ;
        }
}
```

JUNAID. GIRKAR
60004190057
JAGirkar

```
struct node * pop()
{
        struct node * temp;
        if (top == -1)
        {
            printf("stack is empty");
        }
        else
        {
            temp = stk[top];
            top --;
        }
        return temp;
}
void inorder (struct node * temp)
{
         if (temp != NULL)
         {
             inorder (temp ->left);
             printf ("% 3c", temp -> data);
             inorder (temp -> right);
         }
}
```

5

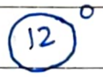
JUNAID . GIRKAR
60004190057
JAGirkar

```
int main ()
{
    char postfix [100];
    int i=0;
    printf("\n Enter the postfix expression :");
    gets(postfix);
    for (i=0; postfix [i] != '\0'; i++)
    {
        struct node * newnode ;
        newnode = (struct node *) malloc (sizeof (struct node));
        if (isalNum(postfix [i]))
        {
            newnode -> data = postfix [i] ;
            newnode -> left = NULL ;
            newnode -> right = NULL ;
            push(newnode);
        }
        else
        {
            struct node * op1 = pop () ;
            struct node * op2 = pop() ;
            newnode -> data = postfix [i] ;
            newnode -> left = op2 ;
            newnode -> right = op1 ;
            push(newnode);
        }
    }
    root = pop();
 root node
    inorder(root);
}
```

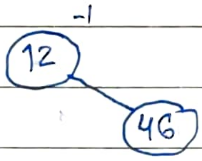
JUNAID. GIRKAR
60004190057
JAGirkar

Q6 To insert : 12, 46, 25, 8, 15, 18, 62, 80, 58

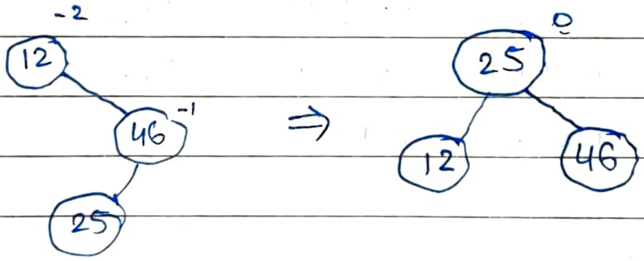
STEP 1 : Insert 12



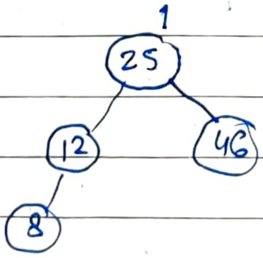
STEP 2 : Insert 46



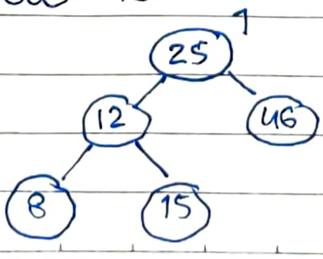
STEP 3 : Insert 25



STEP 4 : Insert 8

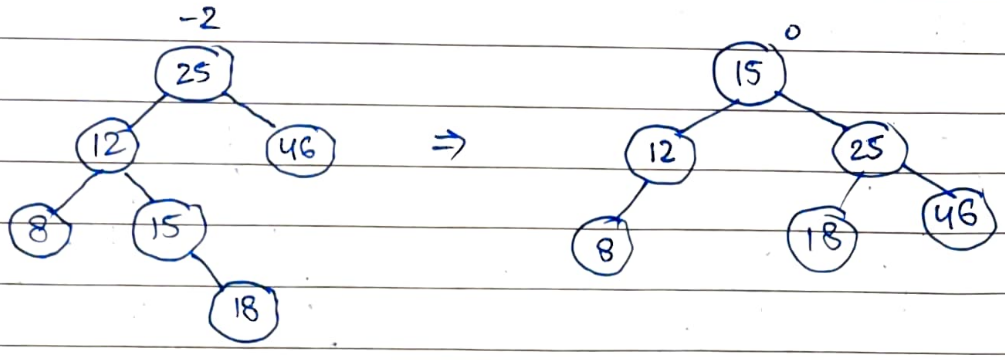


STEP 5 : Insert 15

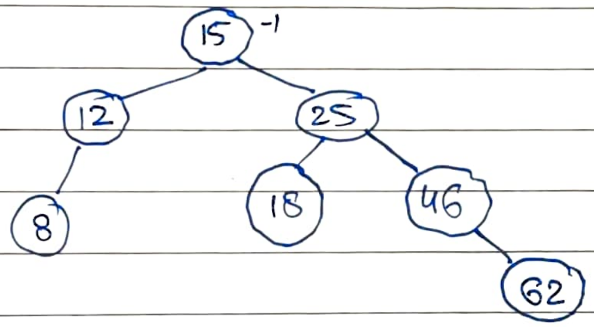


JUNAID. GIRKAR
60004190057
JAGirkar

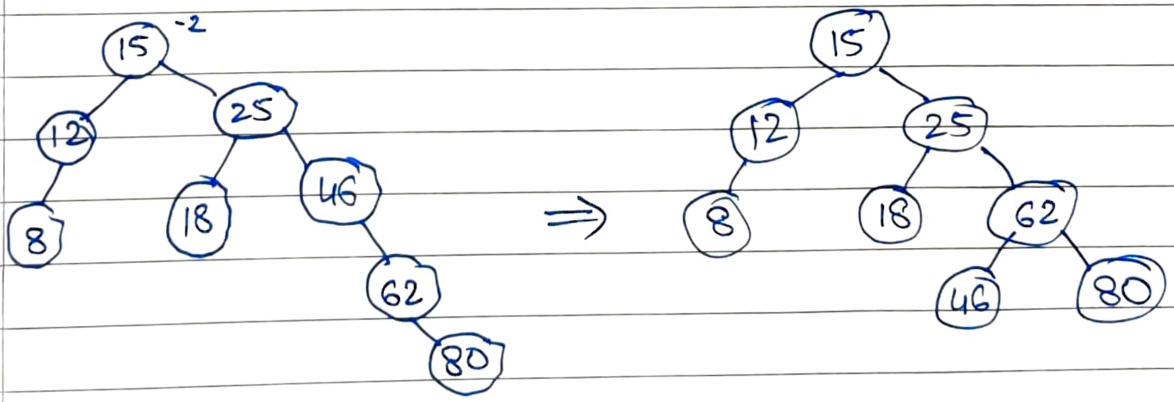
Step 6: Insert 18



STEP 7 : Insert 62

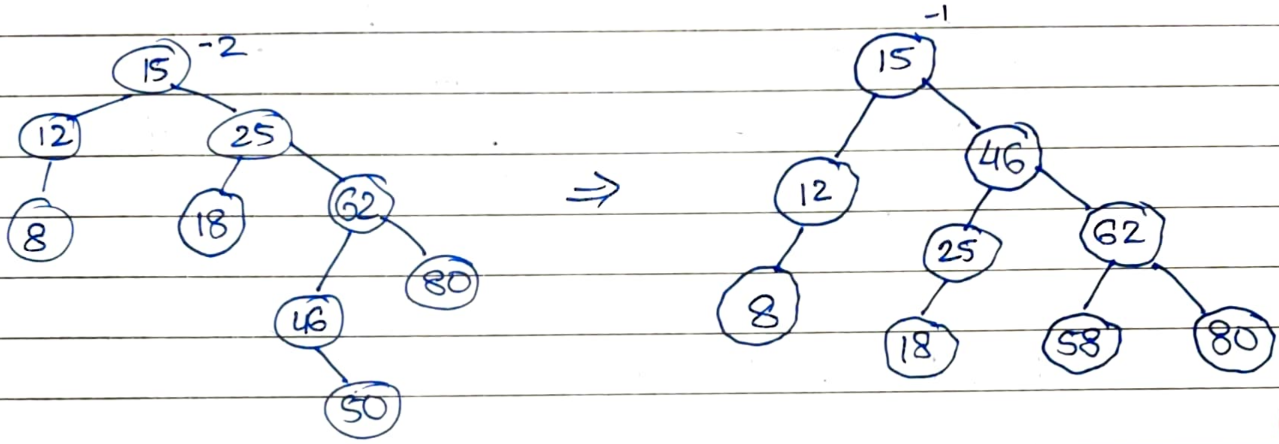


STEP 8 : Insert 80

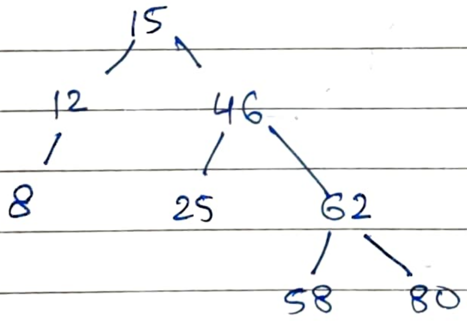


JUNAID · GIRKAR
60004190057
JAGirkar

STEP 9: Insert 58·



ANS



15
12
46
8
25
62
58
80