5.1.1

```c
#include<stdio.h>
#include<stdlib.h>
#include<ctype.h>
#define MAX_SIZE 100
int stack[MAX_SIZE];
int top =-1;
void push(int value){
    if(top<MAX_SIZE-1){
        stack[++top]=value;
    }
}
int pop(){
    int value =-1;
    if(top>=0){
        value=stack[top--];
    }
    return value;
}
int main(){
    char postfix[MAX_SIZE];
    printf("Enter the expression : ");
```

```
    scanf("%s",postfix);
    for(int i=0;postfix[i]!='\0';i++){
        if(isdigit(postfix[i])){
            push(postfix[i]-'0');
        }
        else{
            int operand2=pop();
            int operand1=pop();
            switch(postfix[i]){
                case '+': push(operand1+operand2);break;
                case '-': push(operand1-operand2);break;
                case '*': push(operand1*operand2);break;
                case '/': push(operand1/operand2);break;
            }
        }
    }
    int result = pop();
    printf("The result of expression %s = %d\n",postfix,result);
}
```

5.1.2

```
#include<stdio.h>
void towerOfHanoi(int n,char from_rod,char to_rod,char aux_rod){
    if(n==1)
    {
        printf("Move disk - 1 from pole %c to %c\n",from_rod,to_rod);
        return;
    }
    towerOfHanoi(n-1,from_rod,aux_rod,to_rod);
    printf("Move disk - %d from pole %c to %c\n",n,from_rod,to_rod);
    towerOfHanoi(n-1,aux_rod,to_rod,from_rod);
}

int main(){
    int n;
    printf("Enter number of disks : ");
    scanf("%d",&n);
    towerOfHanoi(n,'A','C','B');
}
```

5.1.3

```c
#include<stdio.h>
int fibonacci(int n){
    if(n==0){
        return 0;
    }
    else if(n==1){
        return 1;
    }
    else{
        return (fibonacci(n-1)+fibonacci(n-2));
    }
}

int main(){
    int n;
    int i;
    printf("Enter the Total terms : ");
    scanf("%d",&n);
    printf("The Fibonacci series of %d terms are : ",n);
    for(i=0;i<n;i++){
        printf("%d ",fibonacci(i));
    }
}

```

5.1.4

```c
#include<stdio.h>
#define N 10
int a[N];
int topa=-1;
int topb=N;
void pusha(){
    int var;
    if(topa==topb-1){
        printf("Stack overflow\n");
        return;
    }
    else{
        printf("Enter data to be pushed: ");
        scanf("%d",&var);

        topa++;
        a[topa]=var;
    }
}
void pushb(int var){
    if(topb-1==topa){
        printf("Stack overflow\n");
    }
    else{
```

```
25          printf("Enter data to be pushed: ");
26          scanf("%d",&var);
27          topb--;
28          a[topb]=var;
29      }
30    }
31  void popa(){
32      int pop;
33      if(topa==-1){
34          printf("Stack 1 is empty\n");
35      }
36      else{
37          pop=a[topa];
38          printf("Popped element: %d\n",pop);
39          topa--;
40      }
41    }
42  void popb(){
43      int pope;
44      if(topb==N){
45          printf("Stack 2 is empty\n");
46      }
47      else{
48          pope=a[topb];
```

```
49          printf("Popped element: %d\n",pope);
50          topb++;
51      }
52    }
53  void display(){
54      int i;
55      printf("Stack 1: ");
56      for(i=topa;i>=0;i--){
57          printf("%d ",a[i]);
58      }
59      printf("\n");
60      printf("Stack 2: ");
61      for(i=topb;i<N;i++){
62          printf("%d ",a[i]);
63      }
64      printf("\n");
65    }
66  int main(){
67      int choice,pos,var;
68      while(1){
69          printf("1. Push\n2. Pop\n3. Display\n4. Quit\n");
70          printf("Enter your choice: ");
71          scanf("%d",&choice);
72          switch(choice){
```

```
73                case 1:
74                    printf("Enter stack number (1 or 2): ");
75                    scanf("%d",&pos);
76
77                    if(pos==1){
78                        pusha(var);
79                    }
80                    else if(pos==2){
81                        pushb(var);
82                    }
83                    break;
84                case 2:
85                    printf("Enter stack number (1 or 2): ");
86                    scanf("%d",&pos);
87                    if(pos==1){
88                        popa();
89                    }
90                    else if(pos==2){
91                        popb();
92                    }
93                    break;
94                case 3:
95                    display();
```

```
96                    break;
97                case 4:
98                    exit(0);
99                default:
100                   printf("Wrong option\n");
101               }
102           }
103       }
```

5.2.1

```c
#include <stdio.h>
#include<stdlib.h>
void print_array(int arr[],int n){
    for(int i=0;i<n;i++){
        printf("%d ",arr[i]);
    }
    printf("\n");
}
void previous_smaller(int arr[],int n){
    int result[n];
    for(int i=0;i<n;i++){
        result[i]=-1;
        for(int j=i-1;j>=0;j--){
            if(arr[j]<arr[i]){
                result[i]=arr[j];
                break;
            }
        }
    }
    print_array(result,n);
}
int main(){
    int arr1[30];
    int n1;
    printf("Enter the size of the array: ");
    scanf("%d",&n1);
    printf("Enter the elements of the array: ");
    for(int i=0;i<n1;i++){
        scanf("%d",&arr1[i]);
    }
    printf("Input: ");
    print_array(arr1,n1);
    printf("Output: ");
    previous_smaller(arr1,n1);
}
```

5.2.2

```c
#include<stdio.h>
#include<string.h>
#define MAX 100
struct Stack{
    int top;
    char items[MAX];
};

void push(struct Stack* stack, char item) {
if(stack->top==MAX-1){
    printf("Stack overflow\n");
}
else{
    stack->top++;
    stack->items[stack->top]=item;
}
    
}

char pop(struct Stack* stack) {
    if(stack->top==-1){
        printf("Stack underflow.\n");
    }
    else{
        char c = stack ->items[stack->top];
        stack->top--;
        return c;
    }
```

```
}
void reverseText(char *t){
    int len=strlen(t),i;
    struct Stack s;
    s.top=-1;
    for(i=len-1;i>=0;i--){
        if(t[i]!=' '){
            push(&s,t[i]);
        }
        else{
            while(s.top!=-1){
                printf("%c",pop(&s));
            }
            printf(" ");
        }
    }
    while(s.top!=-1){
        printf("%c",pop(&s));
    }
}
int main(){
    char t[MAX];
    printf("Enter text: ");
    fgets(t,MAX,stdin);
    t[strcspn(t,"\n")]='\0';
    printf("Reversed text: ");
    reverseText(t);
    printf("\n");
```