DS Assignment 5

1. Write a program to implement a stack and perform basic operations of stack.

1) push 2) pop 3) peek 4) isfull 5) isempty

Source Code:

```
#include<stdio.h>

#include<stdlib.h>

#define SIZE 100

int top=-1,Top=-1;

int stack[SIZE];

char stackchar[SIZE];

void push();

void pushchar();

void pop();

void popchar();

void peek();
```

```
void peekchar();
int isfull(int);
int isempty(int);

void pushchar()
{
	int i;
	char ch;
	if(isfull(Top))
	{
		printf("\nError:Stack is full\n");
	}
	else
	{
		printf("Enter character to be pushed:\n");
		scanf("%s",&ch);
		Top=Top+1;
```

```
        stackchar[Top]=ch;
    }
    if(Top==-1)
    {
        printf("\nStack is empty!!");
    }
    else
    {
        printf("Stack is...\n");
        for(i=Top;i>=0;--i)
            printf("%c ",stackchar[i]);
    }


}
void popchar()
{
  int i;
```

```
    if(isempty(Top))
    {
        printf("\nStack is empty!!");
        return;
    }
    else
    {
        printf("\nDeleted character is
%c\n",stackchar[Top]);
        Top=Top-1;
    }
    if(isempty(Top))
    {
        printf("\nStack is empty!!");
    }
    else
    {
```

```
        printf("\nStack is...\n");
        for(i=Top;i>=0;--i)
            printf("%c ",stackchar[i]);
    }
}
void peekchar()
{
  int i;
    if(isempty(Top))
    {
        printf("Error: Stack not filled\n");
    }
    else
    {
        printf("The last character inserted in the
stack is %c\n", stackchar[Top]);
    }
```

```
}
void push()
{
    int x,i;
    if(isfull(top))
    {
        printf("\nError:Stack is full\n");
    }
    else
    {
        printf("Enter element to push:\n");
        scanf("%d",&x);
        top=top+1;
        stack[top]=x;
    }
    if(top==-1)
    {
```

```
        printf("\nStack is empty!!");
    }
    else
    {
        printf("Stack is...\n");
        for(i=top;i>=0;--i)
            printf("%d ",stack[i]);
    }
}

void pop()
{   int i;
    if(isempty(top))
    {
        printf("\nStack is empty!!");
        return;
    }
```

```
    else
    {
        printf("\nDeleted element is
%d\n",stack[top]);
        top=top-1;
    }
    if(isempty(top))
    {
        printf("\nStack is empty!!");
    }
    else
    {
        printf("Stack is...\n");
        for(i=top;i>=0;--i)
            printf("%d ",stack[i]);
    }
}
```

```
void peek()
{    int i;
      if(isempty(top))
      {
            printf("Error: Stack not filled\n");
      }
      else
      {
            printf("The last Element inserted in the
stack is %d\n", stack[top]);
      }
}
int isfull(int f)
{
      if(f==SIZE-1)
      {
            return 1;
```

```
        }
        else
        {
                return 0;
        }
}
int isempty(int f)
{
        if(f==-1)
        {
                return 1;
        }
        else
        {
                return 0;
        }
}
```

```
int main()
{
	int n;
	printf("\n\nEnter your choice:\n");
	printf("1.Push Integer\n2.Pop Integer\n3.Peek
Integer\n4.Push Character\n5.Pop
Character\n6.Peek Character\n7.Exit\n");
	while(1){
   printf("\nEnter your choice:\n");
	scanf("%d",&n);
	switch(n)
		{
			case 1: push();
				break;
			case 2: pop();
				break;
			case 3: peek();
				break;
```

```
                case 4: pushchar();
                        break;
                case 5: popchar();
                        break;
                case 6: peekchar();
                        break;
                case 7: exit(0);
                default: printf("Enter the correct
choice\n");
            }
        }
        return 0;
}
```

Output:

```
Enter your choice:
1.Push Integer
2.Pop Integer
3.Peek Integer
4.Push Character
5.Pop Character
6.Peek Character
7.Exit
```

```
Enter your choice:
1
Enter element to push:
3
Stack is...
3
Enter your choice:
1
Enter element to push:
4
Stack is...
4 3
Enter your choice:
1
Enter element to push:
5
Stack is...
5 4 3
Enter your choice:
2
Deleted element is 5
Stack is...
4 3
Enter your choice:
3
The last Element inserted in the stack is 4
```

```
Enter your choice:
4
Enter character to be pushed:
a
Stack is...
a
Enter your choice:
```

C:\Users\Dell\Desktop\111.exe

```
a
Enter your choice:
4
Enter character to be pushed:
b
Stack is...
b a
Enter your choice:
4
Enter character to be pushed:
c
Stack is...
c b a
Enter your choice:
5

Deleted character is c

Stack is...
b a
Enter your choice:
6
The last character inserted in the stack is b

Enter your choice:
7

--------------------------------
Process exited after 22.21 seconds with return value 0
Press any key to continue . . .
```

2. Write a program to check string is palindrome using stack.

Source Code:

```
#include<stdio.h>

#include<stdlib.h>

#include<string.h>

#include<math.h>

#define SIZE 100
```

```
int top=-1;
int stack[SIZE];
char push(char);
char pop();
int isfull();
int isempty();
int main()
{
    char s[50],ch;
     int a,count=0,i;
      printf("Enter the string:\n");
       scanf("%s", &s);
       a=strlen(s);
       for(i = 0;s[i]!='\0';i++)
       {
         ch=s[i];
         push(ch);
       }
       for(i=0; i<ceil(a/2); i++)
       {
```

```
            if(stack[i]==stack[a-i-1])
            {
                pop();
                count++;
            }
            else
            {
                printf("Not a Palindome\n");
                return;
            }
        }
        if(count==ceil(a/2))
        {
            printf("It is a palindrome\n");
        }
}
char push(char ch)
{
        if(isfull())
        {
```

```
            printf("\nStack is full!");
        }
        else
        {
            top++;
            stack[top]=ch;
        }
}
char pop()
{
            if(isempty())
            {
                printf("Stack empty\n");
            }
            else
            {
                top--;
            }
}
int isfull()
```

```
{
	if(top==SIZE-1)
	{
		return 1;
	}
	else
	{
		return 0;
	}
}
int isempty()
{
	if(top==-1)
	{
		return 1;
	}
	else
	{
		return 0;
	}
```

}

Output:

```
C:\Users\Dell\Desktop\2.exe
Enter the string:
madam
It is a palindrome

--------------------------------
Process exited after 5.503 seconds with return value 0
Press any key to continue . . .
```

```
C:\Users\Dell\Desktop\2.exe
Enter the string:
choice
Not a Palindome

--------------------------------
Process exited after 4.624 seconds with return value 0
Press any key to continue . . .
```

3. Write a program to sort the string using stack.

Source Code:

```
#include<stdio.h>

#include<string.h>

#define SIZE 100
```

```
char stack1[SIZE];
char stack2[SIZE];
int top=-1,Top=-1;
char push(char,char stack[]);
char pop();
int isfull(char stack[]);
int isempty(char stack[]);

char pop(char stack[])
{
	if(isempty(stack))
	{
		printf("Stack is empty\n");
		return '\0';
	}else
	{
		if(stack==stack1)
		{
			return stack1[top--];
```

```
        }else
        {
            return stack2[Top--];
        }
    }
}
char push(char ch,char stack[])
{
    if(isfull(stack)){
        printf("Stack Overflow\n");
        return '\0';
    }else
    {
        if(stack==stack1)
        {
            top++;
            stack1[top]=ch;
        }else
        {
```

```
                Top++;
                stack2[Top]=ch;
            }
        }
}
int isfull(char stack[])
{
        if(stack==stack1)
        {
            if(top==SIZE-1)
            {
                return 1;
            }else
            {
                return 0;
            }
        }else
        {
            if(Top==SIZE-1)
```

```
            {
                return 1;
            }else
            {
                return 0;
            }
        }
}
int isempty(char stack[])
{
        if(stack==stack1)
        {
            if(top==-1)
            {
                return 1;
            }else
            {
                return 0;
            }
```

```
    }else{
        if(Top==-1)
        {
            return 1;
        }else
        {
            return 0;
        }
    }
}
int main()
{
    char ch[60];
    int i,a;
    printf("Enter the string to be sorted: ");
    scanf("%s",ch);
    push(ch[0],stack1);
    a=strlen(ch);
    i=1;
```

```
while(i<a)
{
    while(!isempty(stack1))
    {
        if((int)stack1[top]>(int)ch[i])
        {
            break;
        }
        else
        {
            push(pop(stack1),stack2);
        }
}
    push(ch[i],stack1);
    while(!isempty(stack2))
    {
        push(pop(stack2),stack1);
    }
    i++;
```

```
    }
    printf("The Sorted string is...\n");
    while(!isempty(stack1)){
        printf("%c",pop(stack1));
    }
}
```

Source Code:

C:\Users\Dell\Desktop\3.exe

```
Enter the string to be sorted: drinking
The Sorted string is...
dgiiknnr
--------------------------------
Process exited after 28.25 seconds with return value 1
Press any key to continue . . .
```