Week 3:

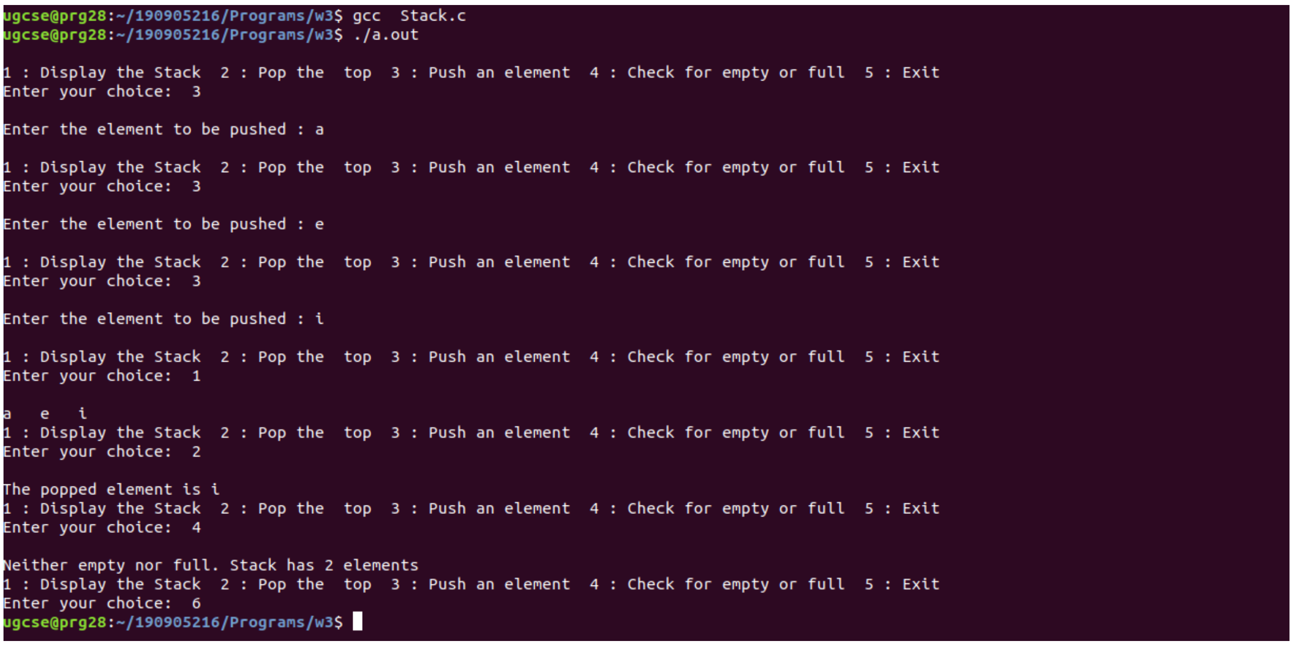
Question 1:

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

#define MAX_STACK_SIZE 50

typedef struct
{
	char array[MAX_STACK_SIZE];
	int top;
} Stack;

int isEmpty(Stack s)
{
	if (s.top == -1)
	{
		return 1;
	}
	return 0;
}

int isFull(Stack s)
{
	if (s.top == MAX_STACK_SIZE - 1)
	{
		return 1;
	}
	return 0;
}

void push(Stack *s, char key)
{
	if (isFull(*s))
	{
		printf("\nThe stack is full");
	}
	s->top++;
	s->array[s->top] = key;
}

char pop(Stack *s)
{
	char temp = s->array[s->top];
	s->top--;
	return temp;
}
```

```
void display(Stack s)
{
	if (isEmpty(s))
	{
		printf("\nThe stack is empty");
	}
	else
	{
		printf("\n");
		for (int i = 0; i <= s.top; i++)
		{
			printf("%c   ", s.array[i]);
		}
	}
}

int main()
{
	Stack charStack;
	charStack.top = -1;
	int choice ;
	char ele;

	do{
		printf("\n1 : Display the Stack  2 : Pop the  top  3 : Push an element  4 : Check for
			empty or full  Any other : Exit");
		printf("\nEnter your choice:  ");
		scanf("%d", &choice);
		getchar();
		switch (choice)
		{
		case 1:
			display(charStack);
			break;

		case 2:
			if (isEmpty(charStack))
			{
				printf("\nThe stack is empty, nothing to pop");
			}
			else
			{
				ele = pop(&charStack);
				printf("\nThe popped element is %c", ele);
			}
			break;

		case 3:
			printf("\nEnter the element to be pushed : ");
			scanf("%c", &ele);
			push(&charStack, ele);
			break;
```

```
                    case 4:
                              if (isEmpty(charStack))
                              {
                                        printf("\nThe stack is empty");
                              }
                              else if (isFull(charStack))
                              {
                                        printf("\nThe stack is full");
                              }
                              else
                              {
                                        printf("\nNeither empty nor full. Stack has %d elements",
                                                  charStack.top + 1);
                              }
                              break;
                    }

          }while(choice<5);
}
```

Output:



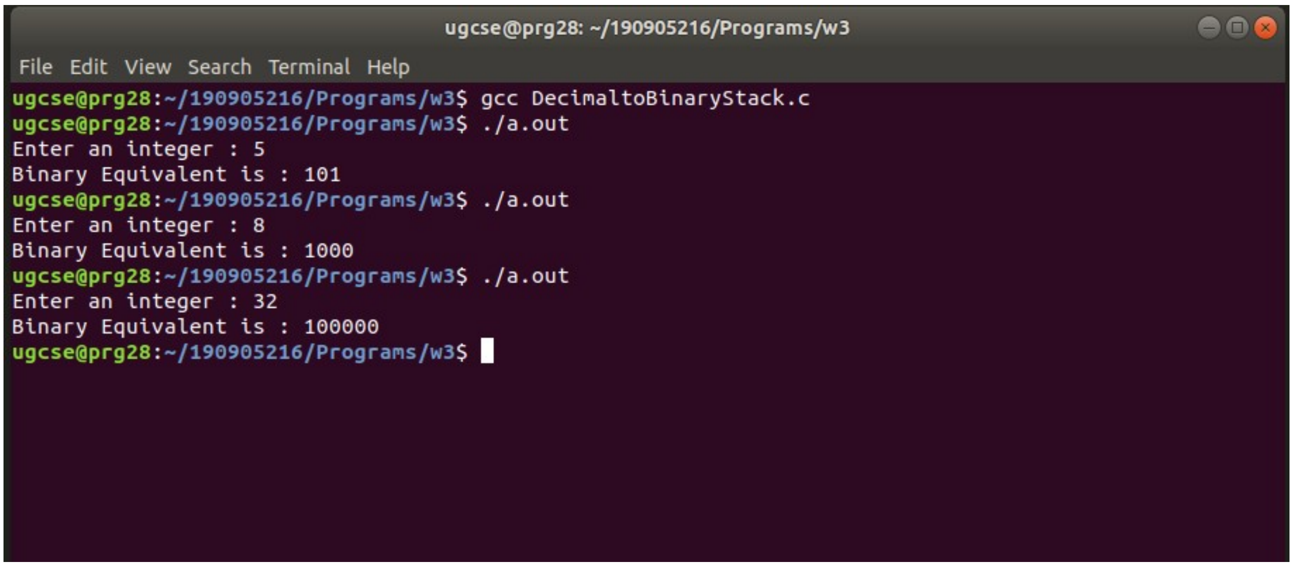
```
ugcse@prg28:~/190905216/Programs/w3$ gcc  Stack.c
ugcse@prg28:~/190905216/Programs/w3$ ./a.out

1 : Display the Stack  2 : Pop the  top  3 : Push an element  4 : Check for empty or full  5 : Exit
Enter your choice:  3

Enter the element to be pushed : a

1 : Display the Stack  2 : Pop the  top  3 : Push an element  4 : Check for empty or full  5 : Exit
Enter your choice:  3

Enter the element to be pushed : e

1 : Display the Stack  2 : Pop the  top  3 : Push an element  4 : Check for empty or full  5 : Exit
Enter your choice:  3

Enter the element to be pushed : i

1 : Display the Stack  2 : Pop the  top  3 : Push an element  4 : Check for empty or full  5 : Exit
Enter your choice:  1

a   e   i
1 : Display the Stack  2 : Pop the  top  3 : Push an element  4 : Check for empty or full  5 : Exit
Enter your choice:  2

The popped element is i
1 : Display the Stack  2 : Pop the  top  3 : Push an element  4 : Check for empty or full  5 : Exit
Enter your choice:  4

Neither empty nor full. Stack has 2 elements
1 : Display the Stack  2 : Pop the  top  3 : Push an element  4 : Check for empty or full  5 : Exit
Enter your choice:  6
ugcse@prg28:~/190905216/Programs/w3$ 
```

```
}
```

Question 2:

```
#include<stdio.h>
#include<stdlib.h>
#define MAX 100


// The remainders are pushed into the stack
void push(int x, int *top, int stack_arr[]){

	if(*top == (MAX-1))
		printf("Stack Overflow!!!!");
	else{
		*top=*top+1;
		stack_arr[*top] = x;
	}
}

//The remainders are popped out of the stack once done, hence they are now in the reverse order
int pop(int *top, int stack_arr[]){

	int x;
	if(*top == -1){

		printf("Stack Underflow!!!!");
		exit(1);
	}
	else{
		x = stack_arr[*top];
		*top=*top-1;
	}
	return x;
}

void DecToBin(int num){

	int stack[MAX], top=-1, rem;
	while(num!=0){

		rem = num%2;
		push(rem, &top, stack);
		num/=2;
	}
	while(top!=-1)
		printf("%d", pop(&top, stack));  //Pops using LIFO, so reverse order of remianders
	printf("\n");
}


int main(){
        int n;
        printf("Enter an integer : ");
        scanf("%d",&n);
        printf("Binary Equivalent is : ");
        DecToBin(n);

        return 0;

}
```

Output:

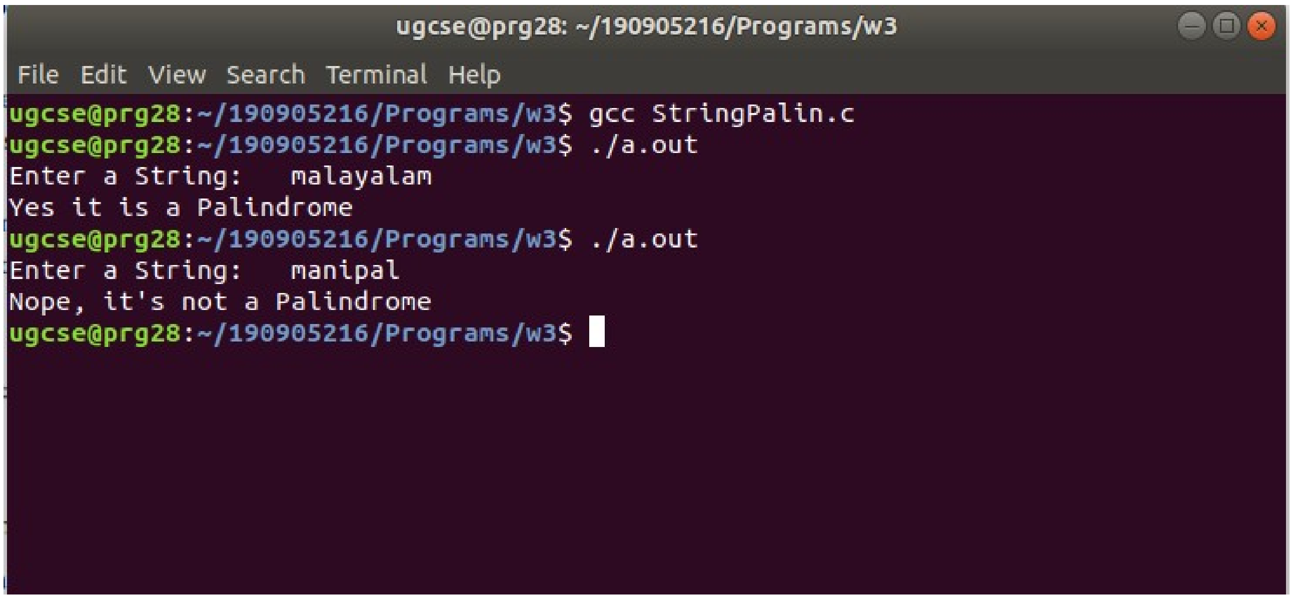

ugcse@prg28: ~/190905216/Programs/w3
File Edit View Search Terminal Help

```
ugcse@prg28:~/190905216/Programs/w3$ gcc DecimaltoBinaryStack.c
ugcse@prg28:~/190905216/Programs/w3$ ./a.out
Enter an integer : 5
Binary Equivalent is : 101
ugcse@prg28:~/190905216/Programs/w3$ ./a.out
Enter an integer : 8
Binary Equivalent is : 1000
ugcse@prg28:~/190905216/Programs/w3$ ./a.out
Enter an integer : 32
Binary Equivalent is : 100000
ugcse@prg28:~/190905216/Programs/w3$
```

Question 3:

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

char* stack;
int top = -1;

void push(char ele) {
    stack[++top] = ele;
}

char pop() {
    return stack[top--];
}

int isPalindrome(char str[])
{
    int length = strlen(str);

    //allocate memory for the stack
    stack = (char*)malloc(length * sizeof(char));

    int i, mid = length / 2;

    //pushing half thr string into the stack
    for (i = 0; i < mid; i++) {
        push(str[i]);
    }

  //ignoring the middle character if string is of odd length
    if (length % 2 != 0) {
        i++;
    }

    //popping it to compare with the rest half of the string
    while (str[i] != '\0') {
        char ele = pop();

        if (ele != str[i])
            return 0; //Not a Palindrome
        i++;
    }

    return 1; //is a Pallindrome
}



int main() {
    char str[100];
    printf("Enter a String:   ");
    scanf("%s",str);

    if (isPalindrome(str)) {
        printf("Yes it is a Palindrome\n");
    }
    else {
        printf("Nope, it's not a Palindrome\n");
    }

    return 0;
}
```

Output:



ugcse@prg28: ~/190905216/Programs/w3
File Edit View Search Terminal Help

```
ugcse@prg28:~/190905216/Programs/w3$ gcc StringPalin.c
ugcse@prg28:~/190905216/Programs/w3$ ./a.out
Enter a String:   malayalam
Yes it is a Palindrome
ugcse@prg28:~/190905216/Programs/w3$ ./a.out
Enter a String:   manipal
Nope, it's not a Palindrome
ugcse@prg28:~/190905216/Programs/w3$ 
```