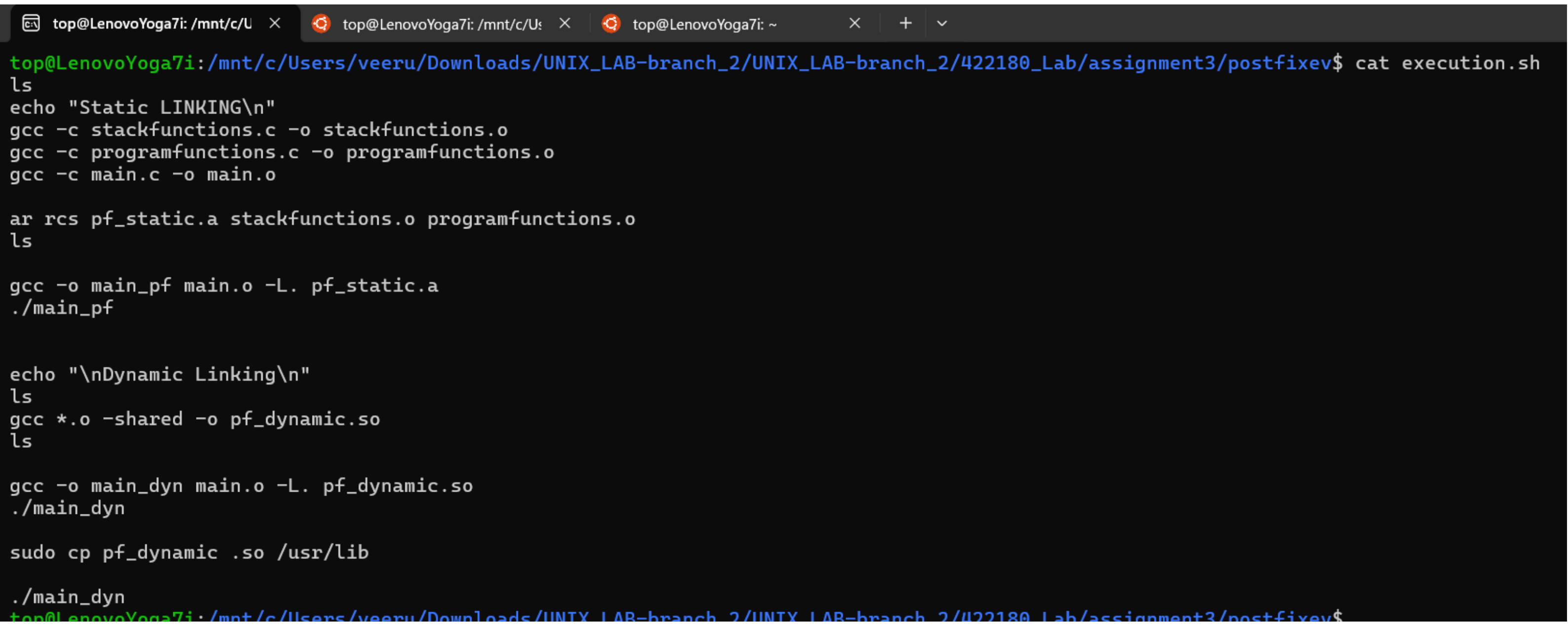
UNIX ASSIGNMENT 3

Name: Veeraja V

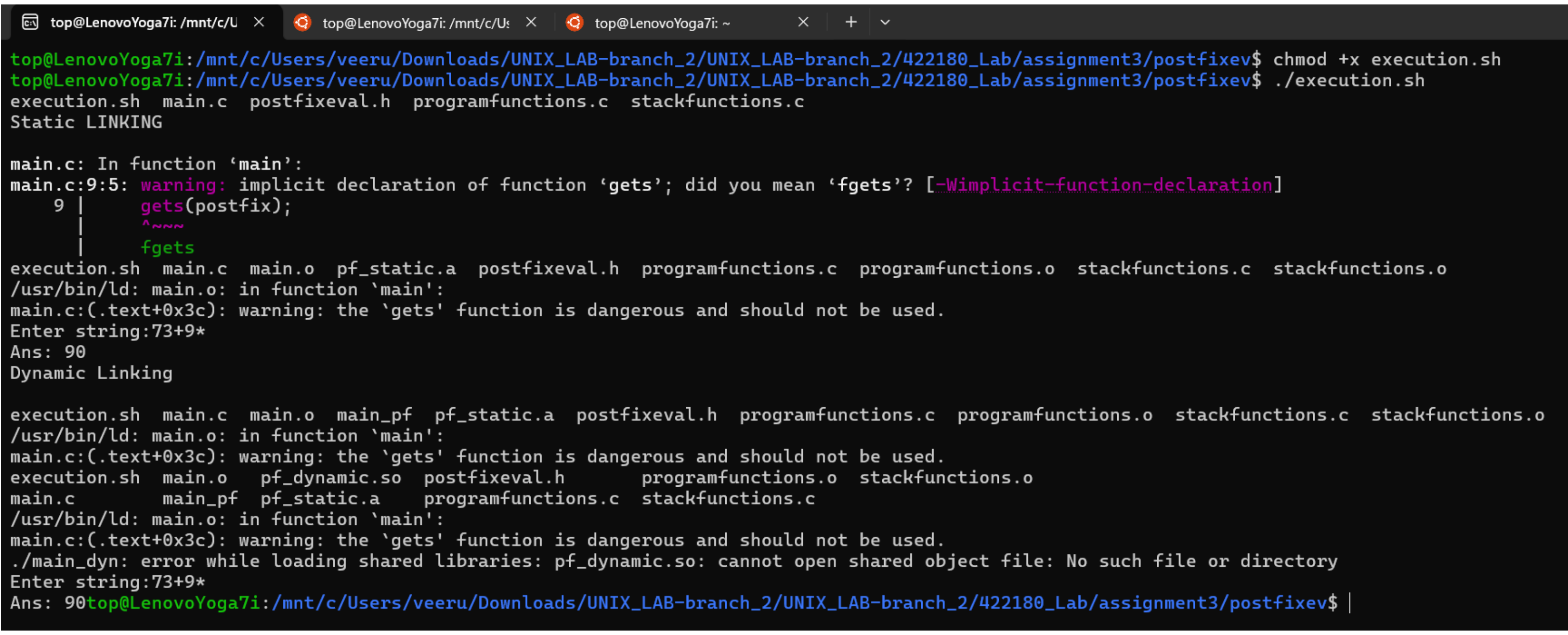
Roll: 422180   Sec: A

Static and Dynamic linking Shell Script:



```
top@LenovoYoga7i:/mnt/c/Users/veeru/Downloads/UNIX_LAB-branch_2/UNIX_LAB-branch_2/422180_Lab/assignment3/postfixev$ cat execution.sh
ls
echo "Static LINKING\n"
gcc -c stackfunctions.c -o stackfunctions.o
gcc -c programfunctions.c -o programfunctions.o
gcc -c main.c -o main.o

ar rcs pf_static.a stackfunctions.o programfunctions.o
ls

gcc -o main_pf main.o -L. pf_static.a
./main_pf


echo "\nDynamic Linking\n"
ls
gcc *.o -shared -o pf_dynamic.so
ls

gcc -o main_dyn main.o -L. pf_dynamic.so
./main_dyn

sudo cp pf_dynamic .so /usr/lib

./main_dyn
top@LenovoYoga7i:/mnt/c/Users/veeru/Downloads/UNIX_LAB-branch_2/UNIX_LAB-branch_2/422180_Lab/assignment3/postfixev$
```

OUTPUT:



```
top@LenovoYoga7i:/mnt/c/Users/veeru/Downloads/UNIX_LAB-branch_2/UNIX_LAB-branch_2/422180_Lab/assignment3/postfixev$ chmod +x execution.sh
top@LenovoYoga7i:/mnt/c/Users/veeru/Downloads/UNIX_LAB-branch_2/UNIX_LAB-branch_2/422180_Lab/assignment3/postfixev$ ./execution.sh
execution.sh  main.c  postfixeval.h  programfunctions.c  stackfunctions.c
Static LINKING

main.c: In function ‘main’:
main.c:9:5: warning: implicit declaration of function ‘gets’; did you mean ‘fgets’? [-Wimplicit-function-declaration]
    9 |     gets(postfix);
      |     ^~~~
      |     fgets
execution.sh  main.c  main.o  pf_static.a  postfixeval.h  programfunctions.c  programfunctions.o  stackfunctions.c  stackfunctions.o
/usr/bin/ld: main.o: in function `main':
main.c:(.text+0x3c): warning: the `gets' function is dangerous and should not be used.
Enter string:73+9*
Ans: 90
Dynamic Linking

execution.sh  main.c  main.o  main_pf  pf_static.a  postfixeval.h  programfunctions.c  programfunctions.o  stackfunctions.c  stackfunctions.o
/usr/bin/ld: main.o: in function `main':
main.c:(.text+0x3c): warning: the `gets' function is dangerous and should not be used.
execution.sh  main.o   pf_dynamic.so  postfixeval.h       programfunctions.o  stackfunctions.o
main.c        main_pf  pf_static.a    programfunctions.c  stackfunctions.c
/usr/bin/ld: main.o: in function `main':
main.c:(.text+0x3c): warning: the `gets' function is dangerous and should not be used.
./main_dyn: error while loading shared libraries: pf_dynamic.so: cannot open shared object file: No such file or directory
Enter string:73+9*
Ans: 90top@LenovoYoga7i:/mnt/c/Users/veeru/Downloads/UNIX_LAB-branch_2/UNIX_LAB-branch_2/422180_Lab/assignment3/postfixev$ |
```

```c
// main.c
#include "postfixeval.h"

int main()
{
    printf("Enter string:");
    char postfix[80];
    gets(postfix);

    int value;
    evaluatepostfix(postfix, &value);
    printf("Ans: %d", value);
    return 0;
}
```

```c
//stackfunctions.c
#include "postfixeval.h"

int init(struct stack *s)
{
    s->top = -1;
    return 1;
}

int isEmpty(struct stack *s)
{
    if (s->top == -1)
        return 1;
    return 0;
}

int isFull(struct stack *s)
{
    if (s->top == SIZE - 1)
        return 1;
    return 0;
}

int push(struct stack *s, int c)
{
    if (isFull(s))
        return 0;

    (s->top)++;
    s->arr[s->top] = c;
    return 1;
}

int pop(struct stack *s, int *c)
{
    if (isEmpty(s))
        return 0;

    *c = s->arr[s->top];
    (s->top)--;
    return 1;
}
```

```
int destroy(struct stack *s)
{
    s->top = -1;
    return 1;
}


//programfunctions.c
#include "postfixeval.h"

int apply(int op1, int op2, char op)
{
    switch (op)
    {
    case '+':
        return op1 + op2;
    case '-':
        return op1 - op2;
    case '*':
        return op1 * op2;
    case '/':
        return op1 / op2;
    }
}

int evaluatepostfix(char *postfix, int *value)
{
    int op1 = 0, op2 = 0;
    int result = 0;
    struct stack s;
    init(&s);

    int i = 0;
    while (postfix[i] != '\0')
    {
        if (postfix[i] >= '0' && postfix[i] <= '9')
        {
            push(&s, postfix[i] - '0');
        }

        else if (postfix[i] == '+' || postfix[i] == '-' || postfix[i]
== '*' || postfix[i] == '/')
        {
            if (isEmpty(&s))
                return 0;
            pop(&s, &op2);

            if (isEmpty(&s))
                return 0;
            pop(&s, &op1);

            result = apply(op1, op2, postfix[i]);
            push(&s, result);
        }
        i++;
    }
    if (!isEmpty(&s))
    {
        pop(&s, value);
        destroy(&s);
        return 1;
    }
    else
    {
        destroy(&s);
        return 1;
    }
    return 0;
}


//postfixeval.h
#include <stdio.h>
#include <string.h>

#define SIZE 100

struct stack
{
    char arr[SIZE];
    int top;
};

//Stack functions
int init(struct stack *s);
int isEmpty(struct stack *s);
int isFull(struct stack *s);
int push(struct stack *s, int c);
int pop(struct stack *s, int *c);
int destroy(struct stack *s);

//Program functions
int apply(int op1, int op2, char op);
int evaluatepostfix(char *postfix, int *value);
```