20IT092 **Shreeram S**

Question 1 Ex.No: 3

AIM :

To perform string operations

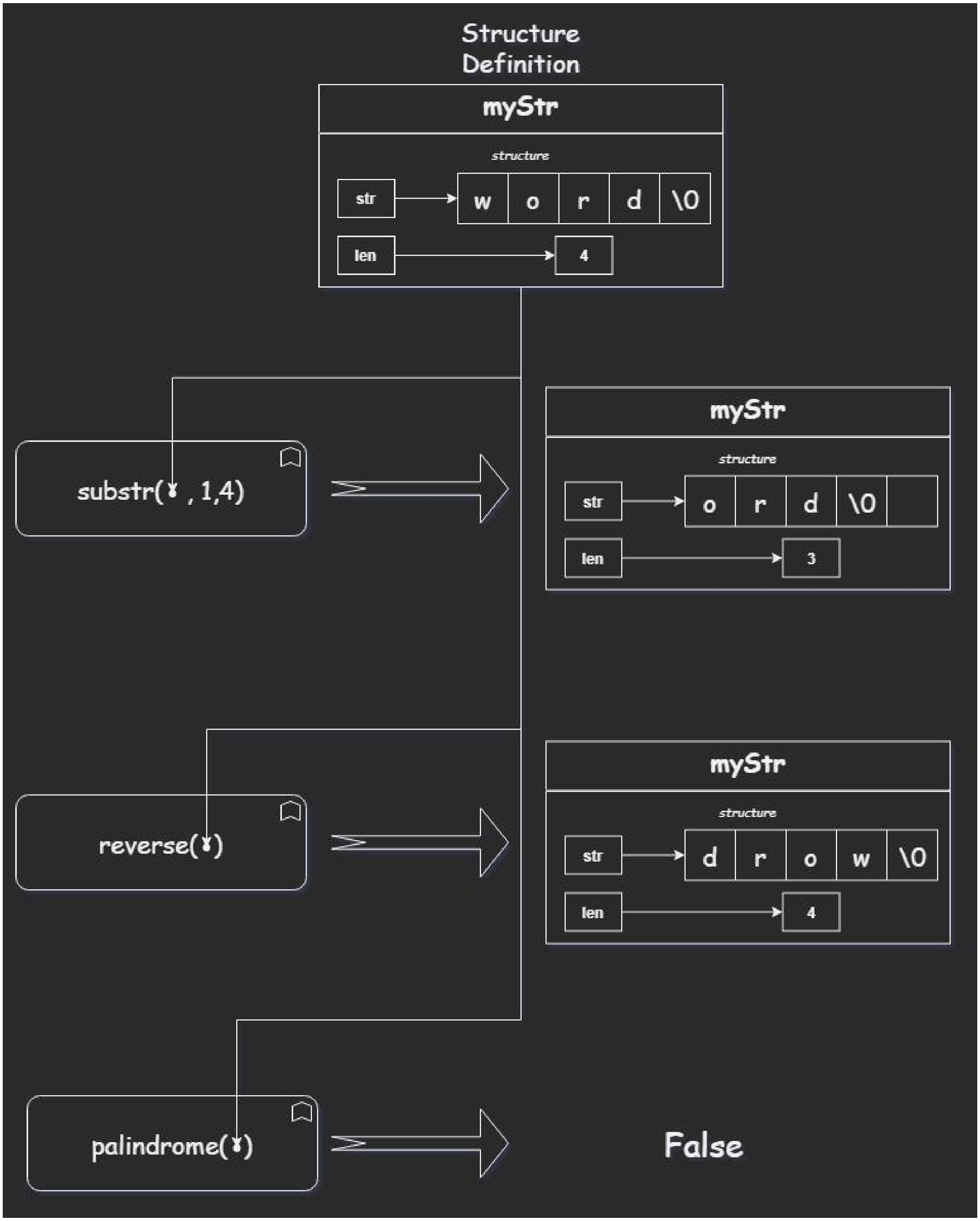
Procedure :

Structure is used for implementation, str is the name of the string, len is the length of the string.

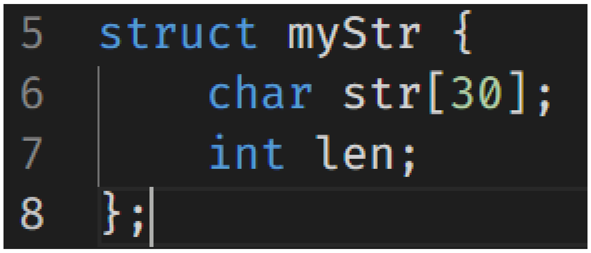
```
reverse(string x):
    begin
    Create new string rev
    Loop in reverse for character in x
        Copy character from str to rev
    Set length of rev as len of str
    Return rev

palindrome(string x):
    begin
    if x is reverse(x)
        return true
    else
        return false

substring(word, start, end):
    begin
    create string sub
    if start < end:
        loop i from start to end:
```

```
        copy word[x] to sub
else:
    loop i from end to start:
        copy word[x] to sub

return sub
```

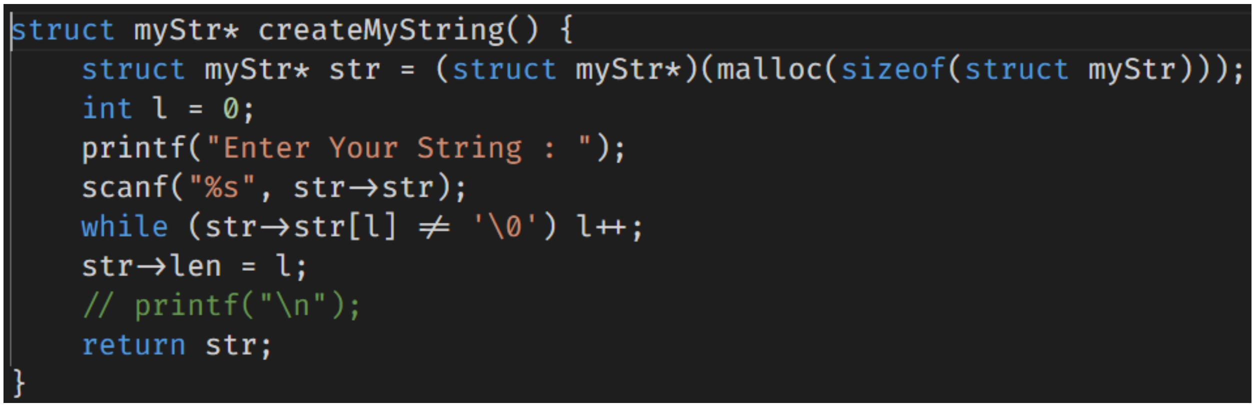
DIAGRAMS :


Structure
Definition
myStr
structure
str
w
o
r
d
\0
len
4
substr(, 1,4)
myStr
structure
str
o
r
d
\0
len
3
reverse( )
myStr
structure
str
d
r
o
w
\0
len
4
palindrome( )
False

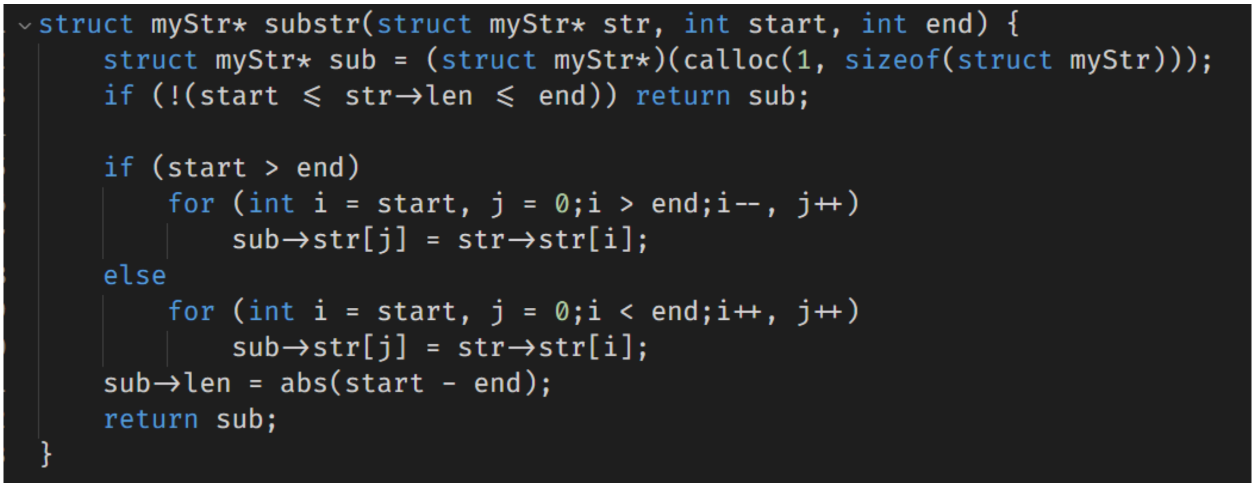
CODE :



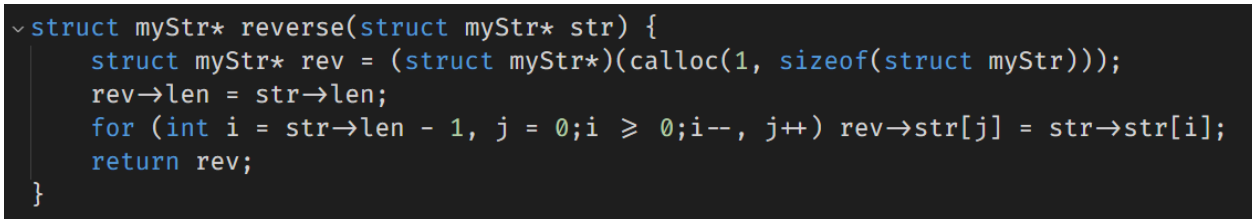
```c
struct myStr {
    char str[30];
    int len;
};
```



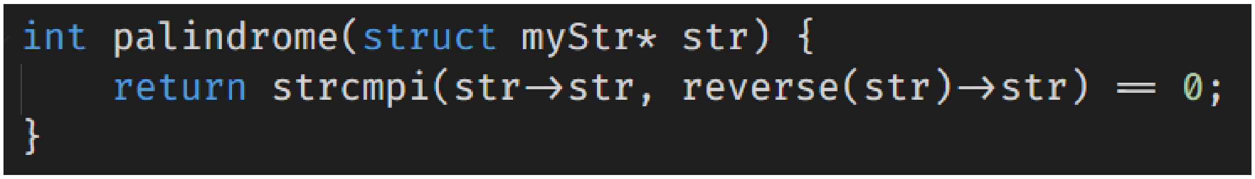
```c
struct myStr* createMyString() {
    struct myStr* str = (struct myStr*)(malloc(sizeof(struct myStr)));
    int l = 0;
    printf("Enter Your String : ");
    scanf("%s", str->str);
    while (str->str[l] != '\0') l++;
    str->len = l;
    // printf("\n");
    return str;
}
```



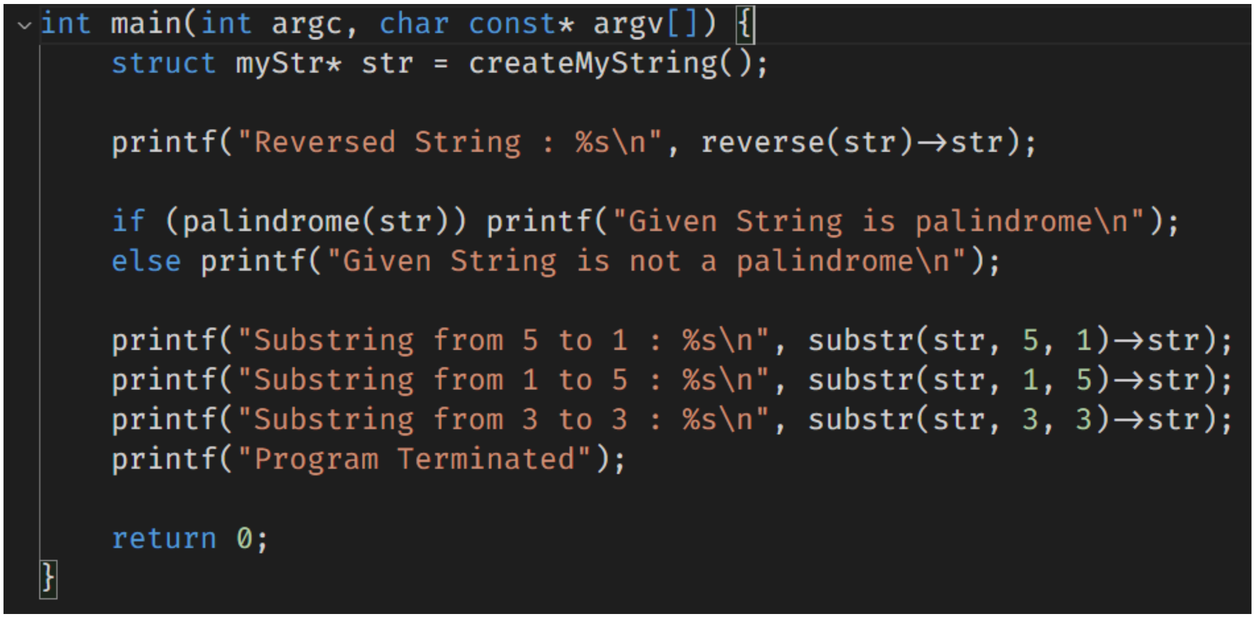
```c
struct myStr* substr(struct myStr* str, int start, int end) {
    struct myStr* sub = (struct myStr*)(calloc(1, sizeof(struct myStr)));
    if (!(start <= str->len <= end)) return sub;

    if (start > end)
        for (int i = start, j = 0;i > end;i--, j++)
            sub->str[j] = str->str[i];
    else
        for (int i = start, j = 0;i < end;i++, j++)
            sub->str[j] = str->str[i];
    sub->len = abs(start - end);
    return sub;
}
```



```c
struct myStr* reverse(struct myStr* str) {
    struct myStr* rev = (struct myStr*)(calloc(1, sizeof(struct myStr)));
    rev->len = str->len;
    for (int i = str->len - 1, j = 0;i >= 0;i--, j++) rev->str[j] = str->str[i];
    return rev;
}
```

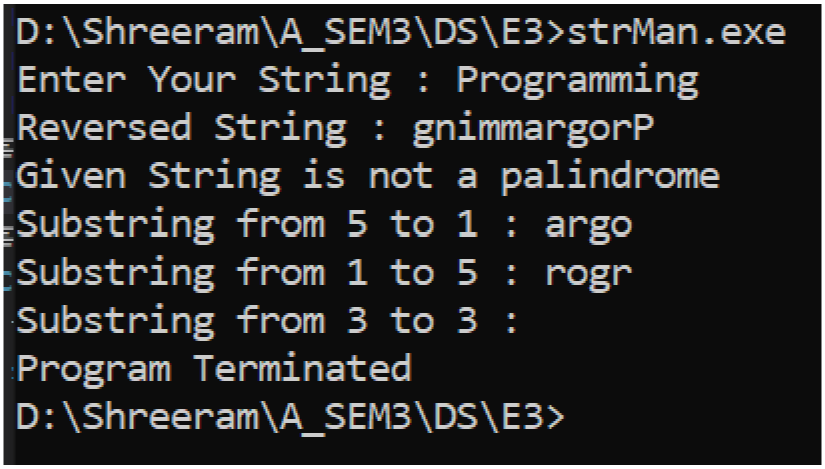


```c
int palindrome(struct myStr* str) {
    return strcmpi(str->str, reverse(str)->str) == 0;
}
```

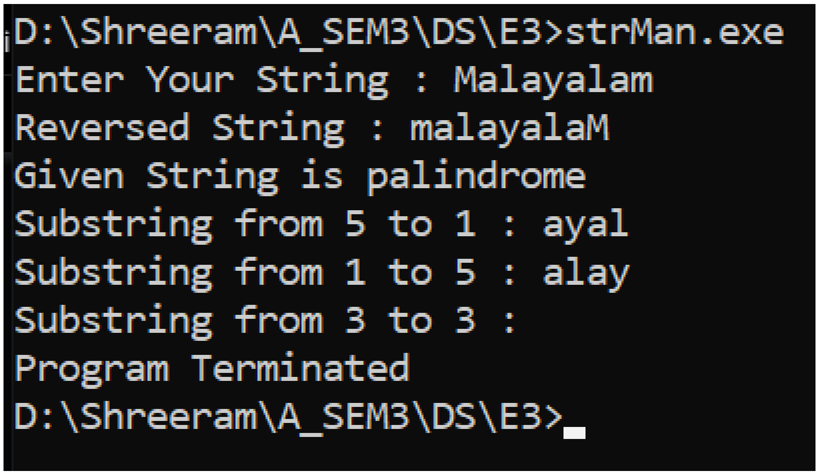


```c
int main(int argc, char const* argv[]) {
    struct myStr* str = createMyString();

    printf("Reversed String : %s\n", reverse(str)->str);

    if (palindrome(str)) printf("Given String is palindrome\n");
    else printf("Given String is not a palindrome\n");

    printf("Substring from 5 to 1 : %s\n", substr(str, 5, 1)->str);
    printf("Substring from 1 to 5 : %s\n", substr(str, 1, 5)->str);
    printf("Substring from 3 to 3 : %s\n", substr(str, 3, 3)->str);
    printf("Program Terminated");

    return 0;
}
```

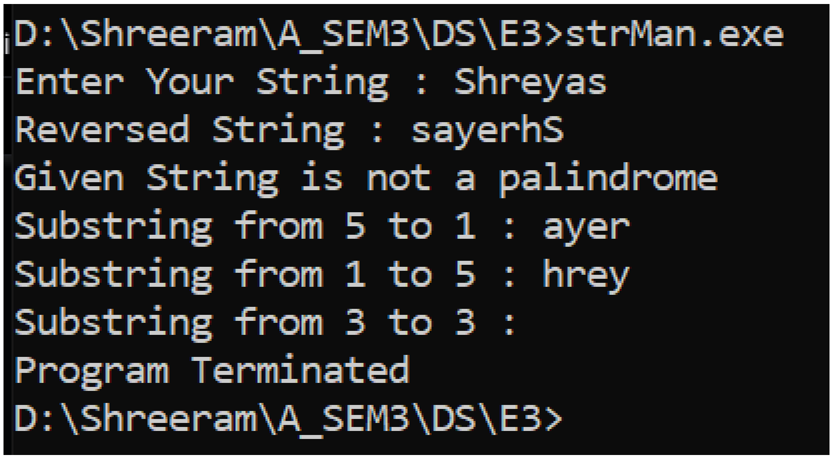
OUTPUT :



```
D:\Shreeram\A_SEM3\DS\E3>strMan.exe
Enter Your String : Programming
Reversed String : gnimmargorP
Given String is not a palindrome
Substring from 5 to 1 : argo
Substring from 1 to 5 : rogr
Substring from 3 to 3 :
Program Terminated
D:\Shreeram\A_SEM3\DS\E3>
```



```
D:\Shreeram\A_SEM3\DS\E3>strMan.exe
Enter Your String : Malayalam
Reversed String : malayalaM
Given String is palindrome
Substring from 5 to 1 : ayal
Substring from 1 to 5 : alay
Substring from 3 to 3 :
Program Terminated
D:\Shreeram\A_SEM3\DS\E3>
```



```
D:\Shreeram\A_SEM3\DS\E3>strMan.exe
Enter Your String : Shreyas
Reversed String : sayerhS
Given String is not a palindrome
Substring from 5 to 1 : ayer
Substring from 1 to 5 : hrey
Substring from 3 to 3 :
Program Terminated
D:\Shreeram\A_SEM3\DS\E3>
```