**String-Programs**

**Strings:**

1. **Declaration and initialization**

#include<stdio.h>

#include<string.h>

int main()

{

char str[]="Hello";//Array of Character

char str1[6]="Hello";

char str2[6]={'H','e','l','l','o','\0'};

char str5[4]={{'B'},{'Y'},{'E'},{'\0'}};

char \*str3="Hello";//Character pointer

char \*abc;

abc="Hello World";// For character pointer it is allowed

//abc=str;//It will work

char str4[5];

//str4="Hello";//Not allowed in C

int x;

printf("str is %s",str);//Value of str

printf("\nstr1 is %s",str1);//value of str1

printf("\nstr2 is %s",str2);//Value of str2

printf("\nstr3 is %s",str3);//Value of str3

printf("\nstr5 is %s",str5);

printf("\nBase address of str string is:%u",str);//Base address of str

printf("\nAddress of first character of abc string is:%u",abc);//Address of first character

abc++;//Pointer incremented by 1

printf("\nAddress of second character of abc string is:%u",abc);//Address of second character

x=strlen(str);

printf("\n%d is the length of str",x);//In calculating length null character is not counted

printf("\nSize of str is:%d",sizeof(str));//In displaying total number of bytes null character is calculated

printf("\nSize of abc is:%d",sizeof(abc));//size will be 8 for pointer variable

return 0;

}

1. **WAP to traverse a string character by character**

#include <stdio.h>

int main()

{

char name[]="Hello World"; //string char array

int i=0;

while(name[i]!='\0') //untill null character

{

printf("%c\n", name[i]);

i++;

}//end while

}//

1. **WAP to replace all spaces in a given string with ‘$’[Example for character replacement]**

#include<stdio.h>

int main()

{

char x[100];

int i=0;

printf("\n Enter the string:");

gets(x);

while(x[i]!='\0')

{

if(x[i]==' ')

{

x[i]='$';//Character replacement

}

i++;

}

printf("\n String after character replacement is:%s",x);

return 0;

}

1. **WAP to find the length of a given string without using strlen()/ or string inbuilt function**

#include<stdio.h>

int main()

{

char x[100];

int i=0;

printf("\n Enter String:");

gets(x);

while(x[i]!='\0')

{

i++;

}

printf("\n Length of the string is:%d",i);

return 0;

}

1. **WAP to copy one string into another string without using strcpy()/ or inbuilt function**

#include<stdio.h>

int main() {

char s1[100], s2[100];

int i;

printf("\nEnter the string :");

gets(s1);//Hello

i = 0;

while (s1[i] != '\0') {

s2[i] = s1[i];

i++;

}

s2[i] = '\0';

printf("\nCopied String is %s ", s2);

return (0);

}

1. **WAP to concatenate(or combine) two strings without using strcat()/ or inbuilt function**

#include<stdio.h>

int main()

{

char str1[100],str2[100],str3[200];

int i=0,j=0;

printf("\n Enter the first string:");

gets(str1);

printf("\n Enter the second string:");

gets(str2);

while(str1[i]!='\0')

{

str3[j]=str1[i];

i++;

j++;

}

i=0;

while(str2[i]!='\0')

{

str3[j]=str2[i];

i++;

j++;

}

str3[j]='\0';

printf("\n The concatenated string is:");

puts(str3);

return 0;

}

1. **WAP to compare two strings without using strcmp()/ or inbuilt function**

#include <stdio.h>

#include<string.h>

int main ()

{

// declare variables

char str1 [30], str2 [30];

int i = 0, flag=0 ,length1, length2, length;

// take two string input

printf ("Enter string1:");

gets (str1);

printf ("\nEnter string2:");

gets (str2);

//length of both string

length1 = strlen (str1);

length2 = strlen (str2);

if(length1>length2)

length=length1;

else

length=length2;

while (i<length)

{

if( str1 [i] == str2 [i])

{

i++;

continue;

}

if( str1 [i] < str2 [i])

{

flag = -1;

break;

}

if( str1 [i] > str2 [i])

{

flag = 1;

break;

}

}

if (flag == 0)

printf ("\nBoth strings are equal ");

if(flag == -1)

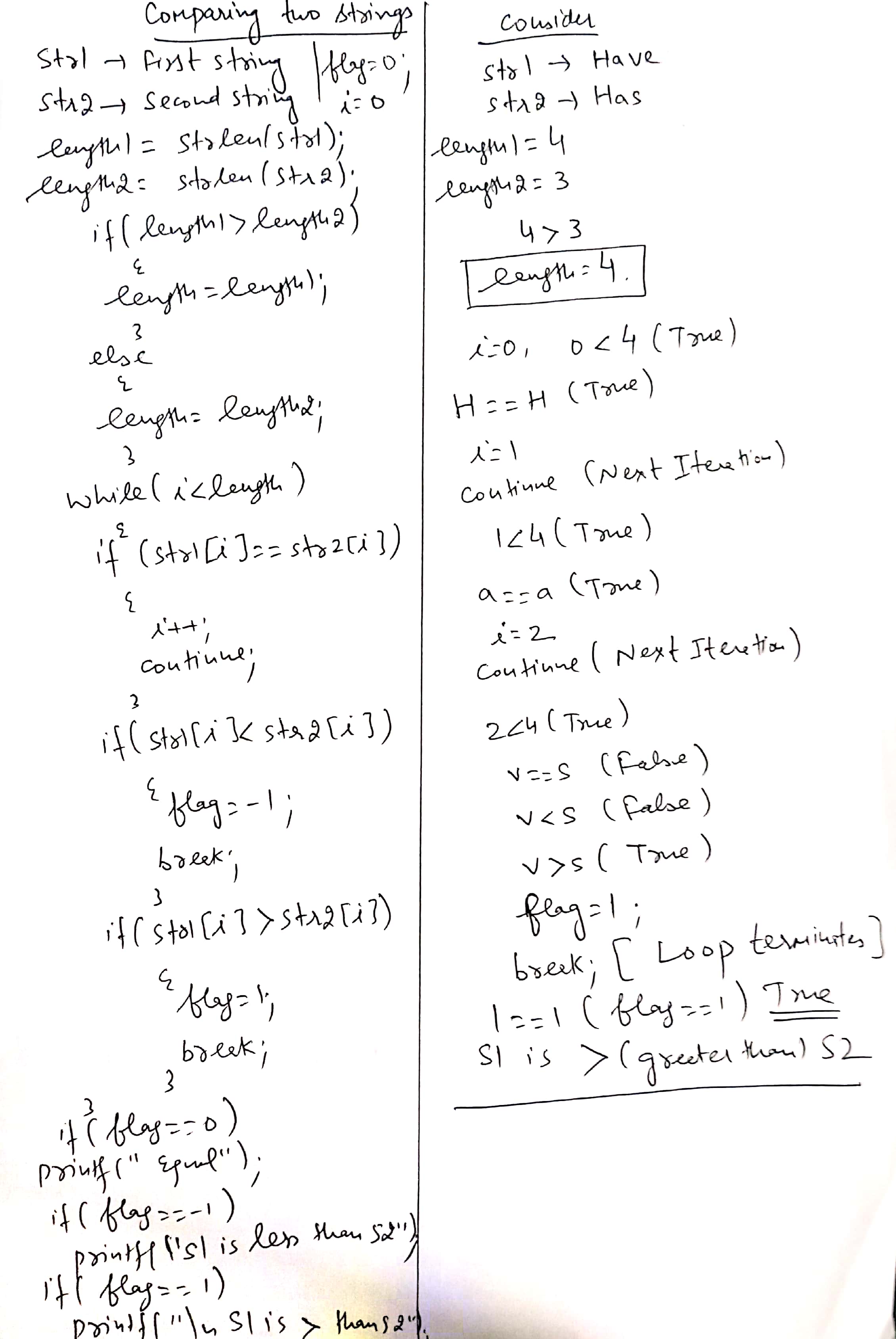
printf ("\nstring1 is less than string2 ");

if( flag == 1)

printf ("\nstring1 is greater than string2 ");

return 0;

}



1. **WAP to display the reverse of a given string without using strrev()/ or inbuilt function**

#include<stdio.h>

#include<string.h>

int main() {

char str[100], temp;

int i, j;

printf("\nEnter the string :");

gets(str);

i = 0;

j = strlen(str) - 1;

while (i < j) {

temp = str[i];

str[i] = str[j];

str[j] = temp;

i++;

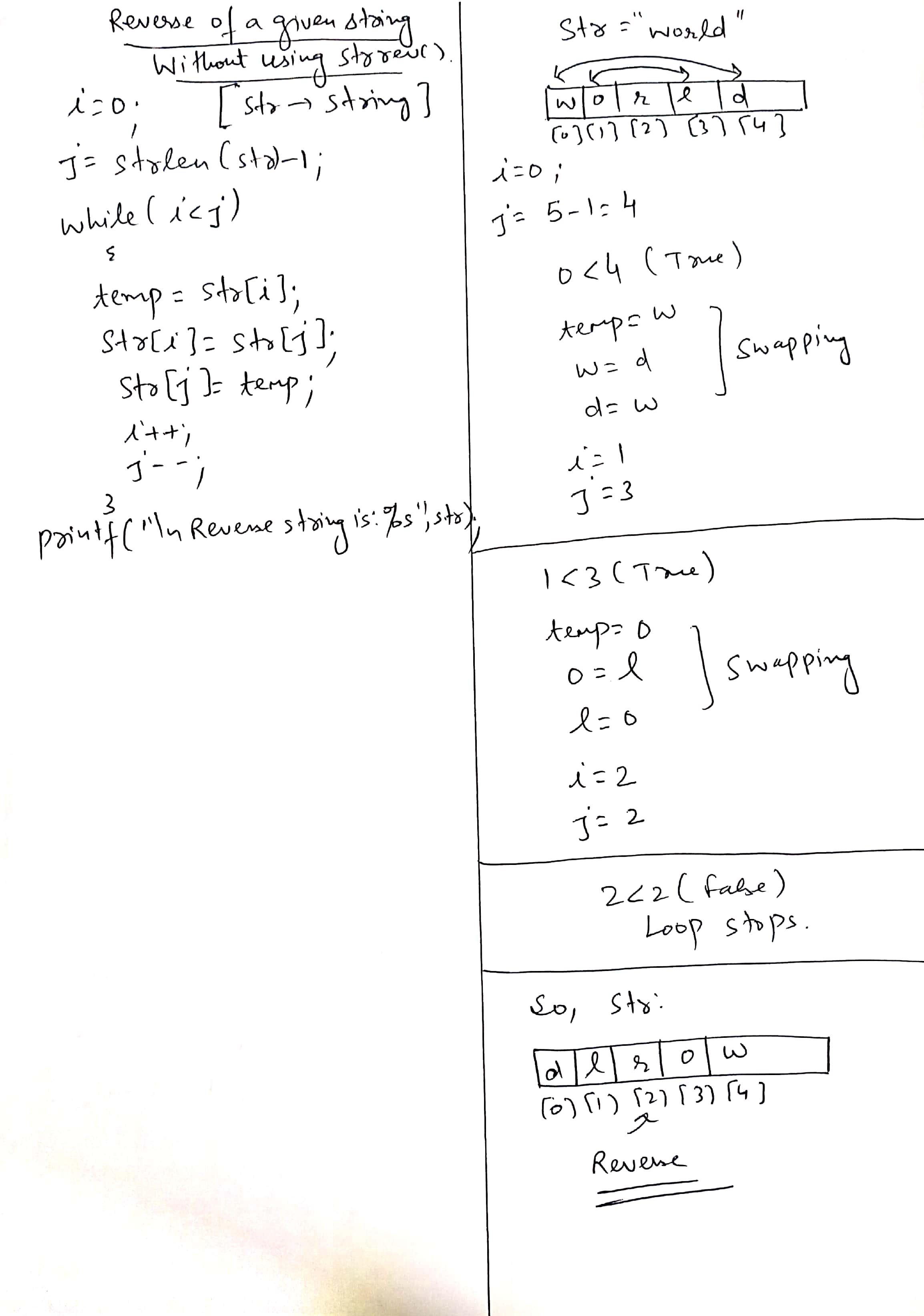
j--;

}

printf("\nReverse string is :%s", str);

return (0);

}



1. **WAP to check whether the given string is palindrome or not(without using strrev())**

#include<stdio.h>

#include<string.h>

int main() {

char str[100], temp;

char str1[100];

int i, j;

printf("\nEnter the string :");

gets(str);

i = 0;

j = strlen(str) - 1;

strcpy(str1,str);

while (i < j)

{

temp = str[i];

str[i] = str[j];

str[j] = temp;

i++;

j--;

}

if(strcmp(str1,str)==0)

{

printf("\n Given String is Palindrome");

}

else

{

printf("\n Not a Palindrome");

}

return (0);

}

1. **WAP to convert all characters of a given string into uppercase without using strupr()/or inbuilt function**

#include<stdio.h>

#include<string.h>

int main()

{

char str1[10];

int i,len;

printf("Enter any string \t");

gets(str1);

len=strlen(str1);

for(i=0;i<len;i++)

{

if(str1[i]>='a' && str1[i]<='z')

str1[i]=str1[i]-32;

}

puts("string in upper is");

puts(str1);

return 0;

}

1. **WAP to convert all characters of a given string into lowercase without using strlwr()/or inbuilt function**

//WAP to convert all uppercase characters of string into lowercase

#include<stdio.h>

#include<string.h>

int main()

{

char str1[10];

int i,len;

printf("Enter any string \t");

gets(str1);

len=strlen(str1);

for(i=0;i<len;i++)

{

if(str1[i]>='A' && str1[i]<='Z')

str1[i]=str1[i]+32;

}

puts("string in lower is");

puts(str1);

return 0;

}

1. **WAP to sort the characters of a given string into ascending order**

#include<stdio.h>

#include<string.h>

int main()

{

char s[10],t;

int n,i,j;

printf("\n Enter String:");

gets(s);

n=strlen(s);

for(i=0;i<n-1;i++)

{

for(j=0;j<n-i-1;j++)

{

if(s[j]>s[j+1])

{

t=s[j];

s[j]=s[j+1];

s[j+1]=t;

}

}

}

printf("%s",s);

}

1. **WAP to count vowels in a given string**

#include<stdio.h>

int main()

{

char x[100];

int i=0,count=0;

printf("\n Enter the string:");

gets(x);

while(x[i]!='\0')

{

if(x[i]=='a'||x[i]=='e'||x[i]=='i'||x[i]=='o'||x[i]=='u'||x[i]=='A'||x[i]=='E'||x[i]=='I'||x[i]=='O'||x[i]=='U')

{

count++;

}

i++;

}

printf("\n Number of vowels in the string are:%d",count);

return 0;

}

1. **WAP to traverse all characters of a given string using pointer to character**

#include<stdio.h>

int main()

{

char \*g="C Programming";

int length=0,i=0;

while(\*g!='\0')

{

printf("%c",\*g);//Value at address

g++;//Pointer is incremented by 1 after each iteration

length++;//Variable for counting length

}

printf("\nLength of the string is:%d",length);

return 0;

}

1. **WAP to count total no. of characters and words in a given string**

#include<stdio.h>

int main()

{

char x[100];

int i=0,length=0,c=0,w=1;

printf("\n Enter String:");

gets(x);

while(x[i]!='\0')

{

if(x[i]==' ' && x[i+1]!=' ')

{

w++;

}

c++;

i++;

}

printf("\n Total number of characters are:%d, and no. of words are:%d",c,w);

return 0;

}

1. **WAP to demonstrate array of strings in C**

#include<stdio.h>

int main()

{

char names[5][10];

int i,n;

printf("\n Enter the number of students:");

scanf("%d",&n);

fflush(stdin);

for(i=0;i<n;i++)

{

printf("\n Enter the name of student %d: ",i+1);

gets(names[i]);

}

printf("\n Names of the students are:\n");

for(i=0;i<n;i++)

puts(names[i]);

return 0;

}

1. **WAP to demonstrate all string inbuilt functions**

#include<stdio.h>

#include<string.h>

int main()

{

char str1[6]="Hello";

char str3[6],str4[2];

char str5[15]="Strings",str6[6]="Class";

char str7[3]="WE",str8[4]="are";

char str9[4]="HOW",str10[4]="how";

char str11[12]="Programming",str12[9]="Programs";

int x;

strcpy(str3,str1);

puts(str3);

strncpy(str4,str1,1);

str4[1]='\0';

puts(str4);

strrev(str3);

puts(str3);

strcat(str5,str6);

puts(str5);

strncat(str5,str6,2);

puts(str5);

strlwr(str7);

puts(str7);

strupr(str8);

puts(str8);

x=strcmp(str9,str10);

printf("\n%d",x);

x=stricmp(str9,str10);

printf("\n%d",x);

x=strncmp(str11,str12,7);

printf("\n%d",x);

printf("\n%d",strlen(str9));

return 0;

}

**Practice questions:**

WAP to replace all vowels in a given string with ‘@’ and display the final string

WAP to count and display the frequencies of characters: ‘p’ and‘s’ in a given string

WAP to take input for a string in uppercase, convert characters ‘h’, and ‘r’ into lowercase, rest should remain as it is.