

Pointers

1. Write a C program to swap two numbers without using third variable using call by reference.

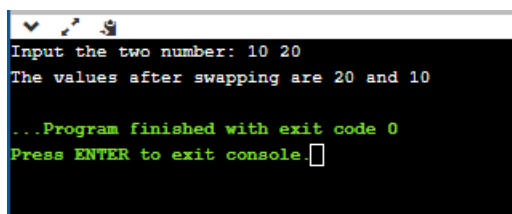
Program:

```
#include <stdio.h>

void swap(int *a,int *b)
{
    *a=*a+*b;
    *b=*a-*b;
    *a=*a-*b;
}

int main()
{
    int n1,n2;
    printf("Input the two number: ");
    scanf("%d%d",&n1,&n2);
    swap(&n1,&n2);
    printf("The values after swapping are %d and %d",n1,n2);
    return 0;
}
```

OUTPUT:



```
Input the two number: 10 20
The values after swapping are 20 and 10
...Program finished with exit code 0
Press ENTER to exit console.
```

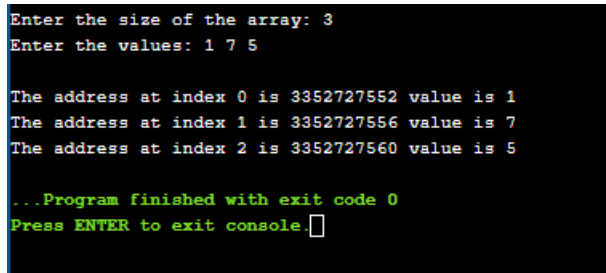
2. Write a C program to read array elements using pointers and print with addresses.

Program:

```
#include <stdio.h>

int main()
{
    int n,i;
    printf("Enter the size of the array: ");
    scanf("%d",&n);
    int arr[n];
    printf("Enter the values: ");
    for(i=0;i<n;i++)
        scanf("%d",(arr+i));
    for(i=0;i<n;i++)
    {
        printf("\nThe address at index %d is %u value is %d",i,arr+i,*(arr+i));
    }
    return 0;
}
```

OUTPUT:

A screenshot of a terminal window showing the execution of the C program. The user enters '3' for the array size and '1 7 5' for the values. The program outputs the memory address and value for each index. The terminal text is as follows:

```
Enter the size of the array: 3
Enter the values: 1 7 5

The address at index 0 is 3352727552 value is 1
The address at index 1 is 3352727556 value is 7
The address at index 2 is 3352727560 value is 5

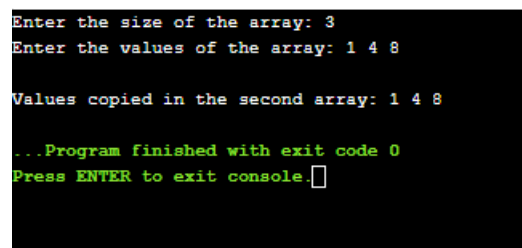
...Program finished with exit code 0
Press ENTER to exit console.
```

3. Write a C program to copy one array to another using pointer.Program:

```
#include <stdio.h>

int main()
{
    int n,i;

    printf("Enter the size of the array: ");
    scanf("%d",&n);
    int arr1[n],arr2[n];
    printf("Enter the values of the array: ");
    for(i=0;i<n;i++)
        scanf("%d",arr1+i);
    for(i=0;i<n;i++)
    {
        *(arr2+i)=*(arr1+i);
    }
    printf("\nValues copied in the second array: ");
    for(i=0;i<n;i++)
        printf("%d ",arr2[i]);
    return 0;
}
```

OUTPUT:A screenshot of a terminal window showing the execution of the C program. The text is as follows:

```
Enter the size of the array: 3
Enter the values of the array: 1 4 8
Values copied in the second array: 1 4 8
...Program finished with exit code 0
Press ENTER to exit console.
```

4. Use an array of function pointers to perform any one of the following arithmetic operation +, -, *, / based on the user choice.

Program:

```
#include <stdio.h>

float add(float a, float b)
{
    return a+b;
}

float sub(float a, float b)
{
    return a-b;
}

float mul(float a, float b)
{
    return a*b;
}

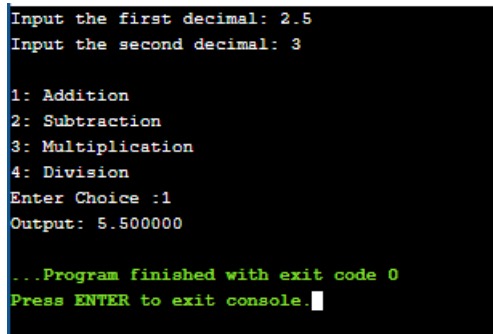
float divi(float a, float b)
{
    return a/b;
}

float (*ptr[4]) (float x, float y);

int main()
{
    float n1,n2;
    int choice;
    printf("Input the first decimal: ");
```

```
scanf("%f",&n1);
printf("Input the second decimal: ");
scanf("%f",&n2);
ptr[0]=add;
ptr[1]=sub;
ptr[2]=mul;
ptr[3]=divi;
printf("\n1: Addition\n2: Subtraction\n3: Multiplication\n4: Division\nEnter
Choice :");
scanf("%d",&choice);
printf("Output: %f",(*ptr[choice-1])(n1,n2));
return 0;
}
```

OUTPUT:



```
Input the first decimal: 2.5
Input the second decimal: 3

1: Addition
2: Subtraction
3: Multiplication
4: Division
Enter Choice :1
Output: 5.500000

...Program finished with exit code 0
Press ENTER to exit console.
```

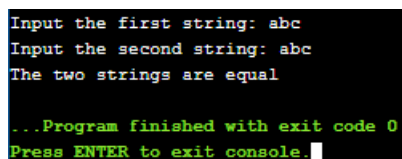
String

Q1. Write a C Program to get two string inputs from the user. Compare the two strings using string function and print strings are equal or not.

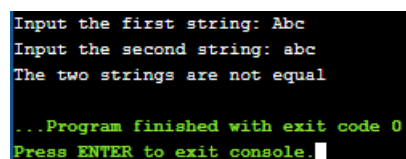
Program:

```
#include <stdio.h>
#include<string.h>
int main()
{
    char str1[50],str2[50];
    printf("Input the first string: ");
    gets(str1);
    printf("Input the second string: ");
    gets(str2);
    if(strcmp(str1,str2))
        printf("The two strings are not equal");
    else
        printf("The two strings are equal");
    return 0;
}
```

OUTPUT:

A screenshot of a terminal window showing the execution of a C program. The user has entered 'abc' for both the first and second strings. The program outputs 'The two strings are equal' and then a message indicating it finished with exit code 0, prompting the user to press ENTER to exit the console.

```
Input the first string: abc
Input the second string: abc
The two strings are equal
...Program finished with exit code 0
Press ENTER to exit console.
```

A screenshot of a terminal window showing the execution of the same C program. The user has entered 'Abc' for the first string and 'abc' for the second string. The program outputs 'The two strings are not equal' and then a message indicating it finished with exit code 0, prompting the user to press ENTER to exit the console.

```
Input the first string: Abc
Input the second string: abc
The two strings are not equal
...Program finished with exit code 0
Press ENTER to exit console.
```

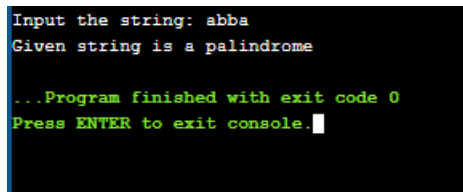
Q2. Write a function `int isPalindrome(char str[])` that takes a string and returns 1 if str is a palindrome and 0 (zero) otherwise. Implement the function by employing a for loop.

Program:

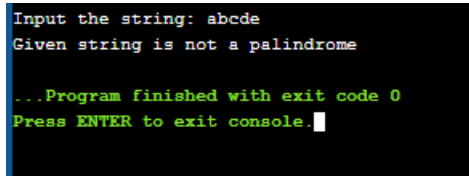
```
#include <stdio.h>
#include<string.h>
int isPalindrome(char str[])
{
    int t=1;
    for(int i=0;i<strlen(str)/2;i++)
    {
        if(str[i]!=str[strlen(str)-i-1])
        {
            t=0;
            break;
        }
    }
    return t;
}
int main()
{
    char c[50];
    printf("Input the string: ");
    gets(c);
    if(isPalindrome(c))
```

```
    printf("Given string is a palindrome");  
else  
    printf("Given string is not a palindrome");  
return 0;  
}
```

OUTPUT:



```
Input the string: abba  
Given string is a palindrome  
...Program finished with exit code 0  
Press ENTER to exit console.
```



```
Input the string: abcde  
Given string is not a palindrome  
...Program finished with exit code 0  
Press ENTER to exit console.
```

Q3. Write a C program to accept first name, middle name and last name from user, store them in 3 different character arrays. Concatenate these 3 strings in one and print as full name.

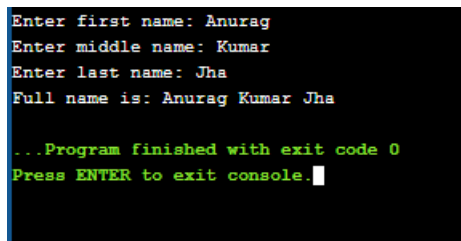
Program:

```
#include <stdio.h>  
  
int main()  
{  
    char c1[50],c2[50],c3[50],c4[150];  
    printf("Enter first name: ");  
    gets(c1);  
    printf("Enter middle name: ");  
    gets(c2);  
    printf("Enter last name: ");  
    gets(c3);  
    strcpy(c4,c1);
```



```
    strcat(c4," ");
    strcat(c4,c2);
    strcat(c4," ");
    strcat(c4,c3);
    printf("Full name is: %s",c4);
    return 0;
}
```

OUTPUT:



```
Enter first name: Anurag
Enter middle name: Kumar
Enter last name: Jha
Full name is: Anurag Kumar Jha

...Program finished with exit code 0
Press ENTER to exit console.
```

Q4. Write a C program to accept a string from the user and count the number of vowels and consonants in it.

Program:

```
#include <stdio.h>
#include<ctype.h>
#include<string.h>
int main()
{
    char str[50];
    int i,aph=0,vow=0,con=0;
    printf("Input the string: ");
    gets(str);
    for(i=0;i<strlen(str);i++)
    {
```

```
        if(isalpha(str[i]))
        {
            aph++;

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

            vow++;

        else

            con++;

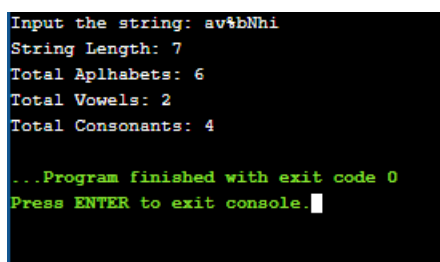
    }
}

printf("String Length: %d",strlen(str));
printf("\nTotal Aplhabets: %d",aph);
printf("\nTotal Vowels: %d",vow);
printf("\nTotal Consonants: %d",con);

return 0;

}
```

OUTPUT:



```
Input the string: avbNhi
String Length: 7
Total Aplhabets: 6
Total Vowels: 2
Total Consonants: 4

...Program finished with exit code 0
Press ENTER to exit console.
```