

ADVANCE

C PROGRAMMING

MODULE - 1

A.BUVANESHKUMAAR

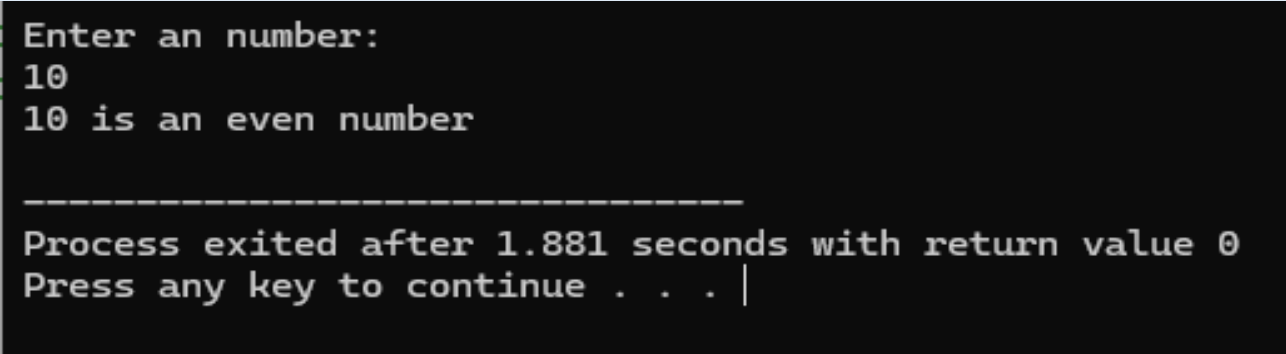
MEPCO SCHLENK ENGINEERING COLLEGE

1. Write a C program to determine the given number is odd or even using Bitwise operators.

Source Code:

```
#include<stdio.h>
#include<stdlib.h>
int main()
{
    int n;
    puts("Enter an number\n");
    scanf("%d",&n);
    if(n & 1)
    {
        printf("%d is an odd number\n",n);
    }
    else
    {
        printf("%d is an even number\n",n);
    }
    return(0);
}
```

Result:

A screenshot of a terminal window showing the execution of a C program. The user enters '10' when prompted 'Enter an number:'. The program outputs '10 is an even number'. Below this, a separator line is shown, followed by the message 'Process exited after 1.881 seconds with return value 0' and 'Press any key to continue . . . |'.

```
Enter an number:
10
10 is an even number

-----
Process exited after 1.881 seconds with return value 0
Press any key to continue . . . |
```

2. Write a C program to count the number of bits set in a number.

Source Code:

```
#include<stdio.h>
#include<stdlib.h>
int main()
{
    int bin_arr[32];int r=0;int value;int temp;int count=0;
    puts("Enter the integer value:");
    scanf("%d",&value);

    while(value>0)
    {
        temp=value%2;
        bin_arr[r]=temp;
        r++;
        value=value/2;
    }
    //len of bin_arr
    int len;
    len=r;
    printf("Binary value for an integer:");
    for(r=len-1;r>=0;r--)
    {

        printf("%d",bin_arr[r]);
    }
    puts("\n");
    //counting for set bits
    for(r=0;r<len;r++)
    {
        if(bin_arr[r]==1)
        {
            count++;
        }
    }
    printf("Number of set bits in an integer value %d is:%d",value,count);
    return(0);
}
```

Result:

```
Enter the integer value:
144
Binary value for an integer:10010000

Number of set bits in an integer value 0 is:2
-----
Process exited after 1.441 seconds with return value 0
Press any key to continue . . . |
```

3. Write a C program to swap two numbers. Use a function pointer to do this operation.

Source Code:

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

void num_swap(int *num1,int *num2)
{
    int temp;//integer declaration
    temp=*num1;
    *num1=*num2;
    *num2=temp;
}

//function pointer
//syntax [ void (*function name)(datatype,datatype);
typedef void (*swap_cond)(int *,int *);

//function
void swapping(swap_cond s_num,int *num1,int *num2)
{
    s_num(num1,num2);
}

int main()
```

```

{
    int a;int b;
    puts("Enter the first number:");
    scanf("%d",&a);
    puts("\n");
    puts("Enter the second number:");
    scanf("%d",&b);

    puts("Before Swapping:");
    printf("num1:%d\t num2:%d",a,b);

    swap_cond s_p=num_swap;
    swapping(s_p,&a,&b);//reference passing

    puts("\n");
    printf("After Swapping:");
    printf("num1:%d\t num2:%d",a,b);
return(0);
}

```

Aliter:

```

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

int swap_num(int *a,int *b)
{
    int temp;
    temp=*a;
    *a=*b;
    *b=temp;
}

int main()
{
    int num1;int num2;
    puts("Enter the number 1:");
    scanf("%d",&num1);

    puts("Enter the number 2:");

```

```

scanf("%d",&num2);

puts("Before Swapping:");
printf("Number 1:%d\tNumber 2:%d",num1,num2);
puts("\n");
//function call
swap_num(&num1,&num2);

puts("After Swapping:");
printf("Number 1:%d\tNumber 2:%d",num1,num2);
return(0);
}

```

Result:

```

Enter the number 1:
10
Enter the number 2:
20
Before Swapping:
Number 1:10      Number 2:20

After Swapping:
Number 1:20      Number 2:10
-----
Process exited after 2.211 seconds with return value 0
Press any key to continue . . . |

```

4. Write an equivalent pointer expression for fetching the value of array element `a[i][j][k][2]`

Source Code:

```

#include <stdio.h>

int main() {
    int a[5][6][7][8] = { { { { 10 } } } }; //initializing 1st element to '10'

    int i=0,j=0,k=0;

```

```

int value1=*(*(*(a+i)+j)+k)+0);

printf("value of a[%d][%d][%d][0]:%d",i,j,k,value1);
puts("\n");

    i = 2, j = 3, k = 4;

int value2 = *(*(*(a + i) + j) + k) + 2);

printf("Value at a[%d][%d][%d][2]: %d\n", i, j, k, value2);

return 0;
}

```

Result:

```

value of a[0][0][0][0]:10
Value at a[2][3][4][2]: 0

-----
Process exited after 0.03159 seconds with return value 0
Press any key to continue . . . |

```

5. Write a C program to Multiply two matrix (n*n) using pointers.

Source Code:

```

#include<stdio.h>
#include<stdlib.h>
#define size 3
int mat_mul(int *a,int *b,int *out,int num)
{
    int r;int c;int k;
    for(r=0;r<num;r++)
    {
        for(c=0;c<num;c++)
        {
            *(out+r*num+c)=0;

```

```

        for(k=0;k<num;k++)
        {

            *(out+r*num+c) += *(a+r*num+k) * *(b+k*num+c);

        }

    }

}

void display_mat(int *res,int num)
{
    int r;int c;

    for(r=0;r<num;r++)
    {
        for(c=0;c<num;c++)
        {
            printf("%d\t",*(res+r*num+c));

        }
        printf("\n");
    }
}

int main()
{
    int mat1[size][size];int mat2 [size][size];int prd_mat[size][size];
    int r1;int r2;
    int row;int col;
    row=size;
    col=size;
    puts("Enter the matrix 1 elements:");
    for(r1=0;r1<row;r1++)
    {
        for(r2=0;r2<col;r2++)
        {
            printf("Enter value for a[%d][%d]:",r1,r2);
            scanf("%d",&mat1[r1][r2]);
        }
    }
}

```



```

puts("\n");
puts("Enter the matrix 2 elements:");
for(r1=0;r1<row;r1++)
{
    for(r2=0;r2<col;r2++)
    {
        printf("Enter value for a[%d][%d]:",r1,r2);
        scanf("%d",&mat2[r1][r2]);
    }
}
//function call
mat_mul(&mat1[0][0],&mat2[0][0],&prd_mat[0][0],size);

//display function call
puts("Resultant Matrix:\n");
display_mat(&prd_mat[0][0],size);

return(0);
}

```

Result:

```

Enter the matrix 1 elements:
Enter value for a[0][0]:2
Enter value for a[0][1]:3
Enter value for a[0][2]:4
Enter value for a[1][0]:5
Enter value for a[1][1]:6
Enter value for a[1][2]:7
Enter value for a[2][0]:8
Enter value for a[2][1]:9
Enter value for a[2][2]:1

Enter the matrix 2 elements:
Enter value for a[0][0]:9
Enter value for a[0][1]:8
Enter value for a[0][2]:7
Enter value for a[1][0]:6
Enter value for a[1][1]:5
Enter value for a[1][2]:4
Enter value for a[2][0]:3
Enter value for a[2][1]:2
Enter value for a[2][2]:1
Resultant Matrix:

48      39      30
102     84      66
129     111     93

-----
Process exited after 35.16 seconds with return value 0
Press any key to continue . . . |

```

6. Find the output of the following

Source Code:

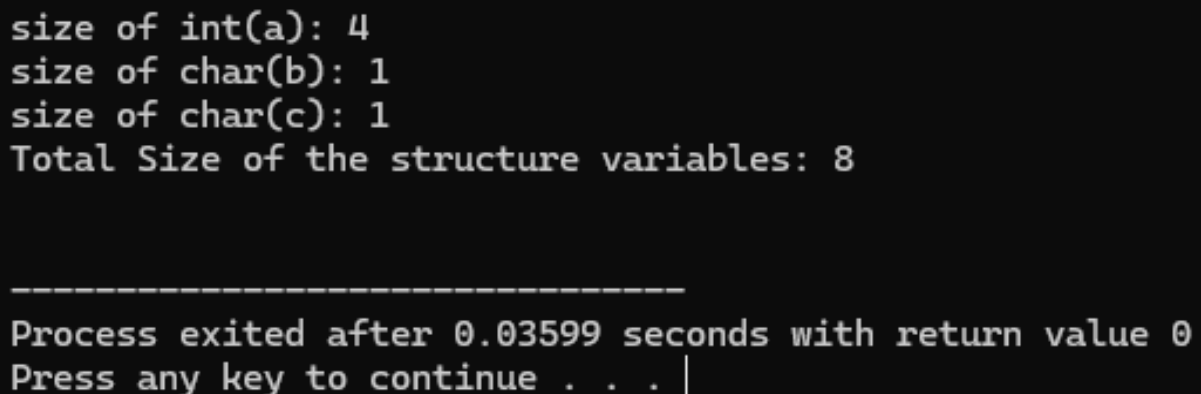
```
#include <stdio.h>

typedef struct
{
    int a;
    char b;
    char c;
} str_var;

int main(int argc, char *argv[])
{
    printf("size of int(a): %d\n", sizeof(((str_var*)0)->a));
    printf("size of char(b): %d\n", sizeof(((str_var*)0)->b));
    printf("size of char(c): %d\n", sizeof(((str_var*)0)->c));
    printf("Total Size of the structure variables: %d\n\n", sizeof(str_var));

    return 0;
}
```

Result://64 bit machine



```
size of int(a): 4
size of char(b): 1
size of char(c): 1
Total Size of the structure variables: 8

-----
Process exited after 0.03599 seconds with return value 0
Press any key to continue . . . |
```

Size of int(a) : 4 + 4(padding)
Size of char(b) : 1
Size of char [c]: 1
Total size of structure variables : 8

//32 bit machine

Size of int(a) : 4

Size of char(b) : 1

Size of char [c]: 1

Total size of structure variables : 6

7. Find the output of the following

Source Code:

```
#include <stdio.h>
typedef struct
{
    char a;
    double b;
    char c;
} str_var;

int main(int argc, char *argv[])
{
    printf("size of char(a): %d\n", sizeof(((str_var*)0)->a));
    printf("size of double(b): %d\n", sizeof(((str_var*)0)->b));
    printf("size of char(c): %d\n", sizeof(((str_var*)0)->c));
    printf("Total Size of the structure variables: %d\n\n", sizeof(str_var));
    return 0;
}
```

Result:

```
size of char(a): 1
size of double(b): 8
size of char(c): 1
Total Size of the structure variables: 24

-----
Process exited after 0.03323 seconds with return value 0
Press any key to continue . . . |
```

//64 bit machine

Size of char [a] : 1 + 7(padding)

Size of double [b] : 8 + 7(padding)

Size of char [c] : 1 + 7(padding)

Total size of structure : 24

//32 bit machine

Size of char [a] : 1 + 3(padding)

Size of double [b] : 8 + 3(padding)

Size of char [c] : 1

Total size of structure : 16

8. Find the output of the following

Source Code:

```
#include<stdio.h>
#include<stdint.h>
int main()
{
    unsigned int var=0x12345678;
    unsigned int rev=0;int r;
    for(r=0;r<8;r++)
    {
        rev=(rev<<4)|((var>>(4*r)) & 0xF);
    }
    printf("%X",rev);
    return(0);
}
```

Result:

87654321

Process exited after 0.03836 seconds with return value 0

Press any key to continue . . . |