

Module 1 Assessment

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

```
#include <stdio.h>
int main()
{
    int num;
    printf("Enter a number:");
    scanf("%d",&num);
    if (num & 1)
    {
        printf("The number is odd");
    }
    else
    {
        printf("The number is even");
    }
    return 0;
}
```

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

```
#include <stdio.h>
int main() {
    int num,bits=0,s;
    printf("Enter the number:");
    scanf("%d",&num);
    while (num > 0)
    {
        s = num&1;
        bits = bits +s;
        num = num>>1;
    }
}
```

```

    }
    printf("Set bits:%d",bits);
    return 0;
}

```

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

```

#include <stdio.h>
int main() {
    int (*ptr) (int*,int*);
    void swap(int* a,int* b)
    {
        int temp;
        temp = *a;
        *a = *b;
        *b = temp;
    }
    ptr = &swap;
    int a = 10, b = 20;
    (*ptr)(&a,&b);
    printf("a = %d\nb = %d",a,b);
    return 0;
}

```

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

Equivalent pointer expression: `*(*(*(*a+i)+j)+k)+2`

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

```

#include <stdio.h>
#include <stdlib.h>
int main()
{
    int n = 3;

```

```

int **mat1 = (int**) malloc(n*sizeof(int));
for (int i = 0; i < n; i++)
{
    mat1[i] = (int*) malloc(n*sizeof(int));
}
int **mat2 = (int**) malloc(n*sizeof(int));
for (int i = 0; i < n; i++)
{
    mat2[i] = (int*) malloc(n*sizeof(int));
}
int **res = (int**) malloc(n*sizeof(int));
for (int i = 0; i < n; i++)
{
    res[i] = (int*) malloc(n*sizeof(int));
}
for (int i = 0; i < n; i++)
{
    for (int j = 0; j < n; j++)
    {
        printf("\nEnter %d%dth element of mat1:",i,j);
        scanf("%d",&mat1[i][j]);
    }
}
for (int i = 0; i < n; i++)
{
    for (int j = 0; j < n; j++)
    {
        printf("\nEnter %d%dth element of mat2:",i,j);
        scanf("%d",&mat2[i][j]);
    }
}
for (int i = 0; i < n; i++)
{
    for (int j = 0; j < n; j++)
    {
        res[i][j] = 0;
        for (int k = 0; k < n; k++)

```

```

        {
            res[i][j] += mat1[i][k]*mat2[k][j];
        }
    }
}
printf("The result matrix:");
for (int i = 0; i < n; i++)
{
    printf("\n");
    for (int j = 0; j < n; j++)
    {
        printf("%d ",res[i][j]);
    }
}
}

```

6. Find the output of the following // Consider the compiler is 32-bit machine

```

#include <stdio.h>
typedef struct
{
    int A;
    char B;
    char C;
} InfoData;
int main(int argc, char *argv[])
{
    //Calculate size of structure
    printf("\n Size of Structure = %d\n\n",sizeof(InfoData));
    return 0;
}

```

Output:

Size of Structure = 8

7. Find the output of the following // Consider the compiler is 32-bit machine

```
#include <stdio.h>
typedef struct
{
    char A;
    double B;
    char C;
} InfoData;
int main(int argc, char *argv[])
{
    //Calculate size of structure
    printf("\n Size of Structure = %d\n\n", sizeof(InfoData));
    return 0;
}
```

Output:

Size of Structure = 16

8. Find the output of the following // Consider the compiler is 32-bit machine

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

Output:

87654321