

```
/*
```

```
-----  
Author- Mahesh Santosh Bhakare  
Employee ID: API2309  
Batch: Greenfield Training (Batch 31)  
-----
```

```
*/
```

```
//Title: 1. To Find whether the given number is prime or not.
```

```
#include <stdio.h>  
int main()  
{  
    int num,i;  
    scanf("%d",&num);  
    if(num==1)  
    {  
        printf("Number is Prime\n");  
    }  
    else  
    {  
        for(i=2;i<=num;i++)  
        {  
            if(num%i==0)  
            {  
                break;  
            }  
        }  
        if(i==num)  
        {  
            printf("Number is Prime\n");  
        }  
        else  
        {  
            printf("Number is not Prime\n");  
        }  
    }  
    return 0;  
}
```

```
//Title: 2. To find whether the given number is even or odd.
```

```
#include <stdio.h>  
int main()  
{  
    int num;  
    scanf("%d",&num);  
    if(num&0 == 0)  
    {  
        printf("Number is Odd\n");  
    }  
    else  
    {  
        printf("Number is Even\n");  
    }  
    return 0;  
}
```

```
//Title: 3. To find whether the given number is divisible by 3 or 5 or 7.
```

```
//Title: 4. Write a program to show if-else statement.
```

```
#include <stdio.h>  
int main()  
{  
    int num,i;  
    scanf("%d",&num);  
    if(num%3 == 0)  
    {
```

```

        printf("Number is Divisible by 3\n");
    }
    if(num%5 == 0)
    {
        printf("Number is Divisible by 5\n");
    }
    if(num%7 == 0)
    {
        printf("Number is Divisible by 7\n");
    }
    else if(num%5 !=0 || num%3 !=0 || num%3 !=0)
    {
        printf("No. not divisible by 3, 5, 7\n");
    }
    return 0;
}

```

//Title: 5. To find the HCF and LCM of any 3 given numbers.

//Title: 6. Write a program to show switch statement.

//Title: 7. Write a program to demonstrate functions.

```

#include <stdio.h>
int min(int a,int b)
{
    if(a<b)
    {
        return a;
    }
    else
    {
        return b;
    }
}
void HCF(int n1, int n2, int n3, int min)
{
    int i;
    for(i=min;i>=1;i++)
    {
        if(n1%i==0 && n2%i==0 && n3%i==0)
        {
            printf("HCF of %d , %d , %d is : %d\n",n1,n2,n3,i);
            break;
        }
    }
}
void LCM(int n1, int n2, int n3, int min)
{
    int i=1;
    while(1)
    {
        if(i%n1==0 && i%n2 ==0 && i%n3 ==0)
        {
            printf("LCM of %d , %d , %d is : %d\n",n1,n2,n3,i);
            break;
        }
        i++;
    }
}
int main()
{
    int n1,n2,n3,choice;
    scanf("%d %d %d",&n1,&n2,&n3);
    do
    {
        printf("1. HCF\n2. LCM \n3. Exit\nEnter your choice: ");
        scanf("%d",&choice);
        switch(choice)
        {
            case 1:

```

```

        HCF(n1,n2,n3,min(min(n1,n2),n3));
        break;
    case 2:
        LCM(n1,n2,n3,min(min(n1,n2),n3));
        break;
    case 3:
        exit(0);
        break;
    default:
        printf("Wrong choice...\n");
}

}while(choice != 3);
return 0;
}

```

//Title: 8. Write a program to implement loops.

```

#include <iostream>
using namespace std;

int main()
{
    int i=0;
    cout<<"While: "<<endl;
    while(i<9)
    {
        cout<<"Hello World"<<endl;
        i++;
    }
    i=0;
    cout<<"Do while: "<<endl;
    do
    {
        cout<<"Hello World"<<endl;
        i++;
    }while(i<9);
    cout<<"For : "<<endl;
    for(i=0;i<9;i++)
    {
        cout<<"Hello World"<<endl;
    }
    return 0;
}

```

//Title: 9. Write a program to demonstrate arrays.

//Title: 10. Write a program to add two matrices.

```

#include <iostream>
using namespace std;

int main()
{
    int i,j,m1,n1,m2,n2,a[10][10],b[10][10],c[10][10];
    cout<<"enter the no. of rows in 1st array: ";
    cin>>m1;
    cout<<"enter the no. of columns in 1st array: ";
    cin>>n1;
    cout<<"enter the elements: "<<endl;
    for(i=0;i<m1;i++)
    {
        for(j=0;j<n1;j++)
        {
            cin>>a[i][j];
        }
    }
    cout<<"enter the no. of rows in 2nd array: ";
    cin>>m2;
    cout<<"enter the no. of columns in 2nd array: ";
}

```

```

cin>>n2;
cout<<"enter the elements: "<<endl;
for(i=0;i<m2;i++)
{
    for(j=0;j<n2;j++)
    {
        cin>>b[i][j];
    }
}
if(m1==m2 && n1==n2)
{
    for(i=0;i<m1;i++)
    {
        for(j=0;j<n1;j++)
        {
            c[i][j]=a[i][j]+b[i][j];
        }
    }

    cout<<"Addition of two matrices is: "<<endl;
    for(i=0;i<m1;i++)
    {
        for(j=0;j<n1;j++)
        {
            cout<<c[i][j]<<" ";
        }
        cout<<endl;
    }
}
else
{
    cout<<"enter the proper array sizes..."<<endl;
}
return 0;
}

```

//Title: 11. Write a program to multiply two matrices.  
//Title: 12. Write a program to demonstrate pointers.

```

#include <iostream>
using namespace std;

int main()
{
    int c1,c2,r1,r2;
    int num;
    int **a=NULL,**b=NULL,**c=NULL;
    cout<<"Enter row and column for matrix 1: "<<endl;
    cin>>r1>>c1;

    a=(int**)malloc(r1*sizeof(int*));
    for(int i=0;i<r1;i++)
    {
        *(a+i)=(int*)malloc(c1*sizeof(int));
        for(int j=0;j<c1;j++)
        {
            cin>>num;
            *((a+i)+j) = num;
        }
    }
    cout<<"Enter row and column for matrix 2: "<<endl;
    cin>>r2>>c2;

    b=(int**)malloc(r2*sizeof(int*));
    for(int i=0;i<r2;i++)
    {
        *(b+i)=(int*)malloc(c2*sizeof(int));
    }
}

```

```

        for(int j=0;j<c2;j++)
        {
            cin>>num;
            (*(b+i)+j) = num;
        }
    }
    if(c1==r2)
    {
        c = (int**)malloc(r1*sizeof(int*));
        for(int i=0;i<r1;i++)
        {
            *(c+i) = (int*)calloc(c2,sizeof(int));
            for(int j=0;j<c2;j++)
            {
                for(int k=0;k<c1;k++)
                {
                    (*(c+i)+j) += (*(a+i)+k)*(*(b+k)+j));
                }
            }
        }
        cout<<"Multiplication Matrix is: "<<endl;
        for(int i=0;i<r1;i++)
        {
            for(int j=0;j<c2;j++)
            {
                cout<<*(*(c+i)+j)<<" ";
            }
            cout<<endl;
        }
    }
    else
    {
        cout<<"Wrong matrix Size"<<endl;
    }
}

```

//Title: 13. Write a program to square a matrix.

```

int main()
{
    int c1,c2,r1,r2;
    int num;
    int **a=NULL,**b=NULL,**c=NULL;
    cout<<"Enter row and column for matrix 1: "<<endl;
    cin>>r1>>c1;

    a=(int**)malloc(r1*sizeof(int*));
    for(int i=0;i<r1;i++)
    {
        *(a+i)=(int*)malloc(c1*sizeof(int));
        for(int j=0;j<c1;j++)
        {
            cin>>num;
            (*(a+i)+j) = num;
        }
    }
    c = (int**)malloc(r1*sizeof(int*));
    for(int i=0;i<r1;i++)
    {
        *(c+i) = (int*)calloc(c2,sizeof(int));
        for(int j=0;j<c1;j++)
        {
            for(int k=0;k<c1;k++)
            {
                (*(c+i)+j) += (*(a+i)+k)*(*(a+k)+j));
            }
        }
    }
}

```

```

    }
}
cout<<"Square Matrix is: "<<endl;
for(int i=0;i<r1;i++)
{
    for(int j=0;j<c2;j++)
    {
        cout<<*((c+i)+j)<<" ";
    }
    cout<<endl;
}
}

```

//Title 14. Write a program to demonstrate type casting.

```

#include <iostream>
using namespace std;

int main()
{
    //implicit casting
    int x = 1;
    char y = 'a';
    x = x + y;
    float z = x + 1.0;
    cout<<"x = " << x << endl;
    cout<<"y = " << y << endl;
    cout<<"z = " << z << endl;
}

```

```

//explicit casting
double a=2.0;
x = (int)a+z;
cout<<x;
return 0;
}

```

// const cast

```

#include <iostream>
using namespace std;

int main()
{
    int const a=10;
    int*p = NULL;
    p = const_cast<int*>(&a);
    (*p)++;
    cout<<a<<endl<<*p;
    return 0;
}

```

// static cast

```

#include<iostream>
class A
{
public:
    int a;
    A()
    {
        a=10;
    }
};
class B
{
public:
    int b;
}

```

```

        B()
        {
            b=20;
        }
    };
    class C
    {
    public:
        int c;
        C()
        {
            c=30;
        }
    };

    class D: public A, public B, public C
    {
    public:
        int d;
        D()
        {
            d =40;
        }
    };

    int main()
    {
        D obj;

        A *pa = &obj;
        B *pb = &obj;
        C *pc = &obj;
        D *pd = &obj;
        pd = static_cast<D*>(pb); //static cast or down casting.

        std::cout<<pd->a<<std::endl;
        std::cout<<pd->b<<std::endl;
        std::cout<<pd->c<<std::endl;
        std::cout<<pd->d<<std::endl;

        return 0;
    }

    // reinterpret cast

#include <iostream>
using namespace std;
class Demo
{
    int a;
    int b;
    int c;
    public:
    Demo(int a, int b, int c)
    {
        this->a = a;
        this->b = b;
        this->c = c;
    }
};
int main()
{
    Demo d(10,20,30);
    int* p=NULL;
    p = reinterpret_cast<int*>(&d);
    cout<<*(p+0)<<endl;
}

```

```

    cout<<*(p+1)<<endl;
    cout<<*(p+2)<<endl;
    return 0;
}

```

//Title 15. Write a program to demonstrate pointers using singly linked list program.

```

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

struct node
{
    int data;
    struct node* next;
};

// ----- FUNCTION TO CREATE NODE -----
struct node* CreateNode()
{
    struct node* newnode = NULL;
    newnode = (struct node*)malloc(sizeof(struct node));
    if(newnode != NULL)
    {
        printf("Enter the data for newnode: ");
        scanf("%d",&(newnode->data));
        newnode->next = NULL;
    }
    else
    {
        printf("Memory not allocated .....\\n");
    }
    return newnode;
}

// ----- FUNCTION TO COUNT NODES -----
int CountNode(struct node* head)
{
    int count = 0;
    while(head != NULL)
    {
        count++;
        head = head->next;
    }
    return count;
}

// ----- FUNCTION TO CREATE LINKED LIST(MAKES LINKING OF NODES) -----
void CreateLinkedList(struct node** head)
{
    struct node* newnode;
    struct node* tempnode = *head;
    newnode = CreateNode();
    if(*head == NULL)
    {
        *head = newnode;
    }
    else
    {
        while(tempnode->next != NULL)
        {
            tempnode = tempnode->next;
        }
        tempnode->next = newnode;
    }
}

```



```

    }
}

// ----- FUNCTION TO DISPLAY LINKED LIST -----
void DisplayLinkedList(struct node* tempnode)
{
    printf("Our Linked List is: ");
    while(tempnode != NULL)
    {
        printf(" -> %d",tempnode->data);
        tempnode = tempnode->next;
    }
    printf("\n");
}

int main()
{
    int choice;
    struct node* first=NULL;
    do
    {
        printf(" ----- *****\n\n");
        printf("1) Create Link List\n2) Display Link List\n3) Exit\nEnter your choice: ");
        scanf("%d",&choice);
        switch(choice)
        {
            case 1: CreateLinkedList(&first); // &first = to make change at the address
of first
                    break;
            case 2: DisplayLinkedList(first); // first = to make copy of first
                    break;

        }
    }while(choice != 0);
    return 0;
}

```

//Title: 16 Write a program to demonstrate strings.

```

#include<bits/stdc++.h>
#include<string.h>

using namespace std;

int main()
{
    char str1[10];
    char str2[10];
    cout<<"Enter String 1: ";
    fgets(str1,sizeof(str1),stdin);
    cout<<"Enter String 2: ";
    fgets(str2,sizeof(str2),stdin);

    cout<<"String Compaire: "<<endl;
    if(strcmp(str1,str2)==0)
    {
        cout<<"equals"<<endl;
    }
    else
    {
        cout<<"not Equals"<<endl;
    }

    cout<<"String Concat: "<<endl;
    strcat(str1,str2);
}

```

```

cout<<str1<<endl;
cout<<str2<<endl;

cout<<"String Copy: "<<endl;
strcpy(str1,str2);
cout<<str1<<endl;
cout<<str2<<endl;

    return 0;
}

```

//Title: 17. Write a program to demonstrate operators in C++.

```

#include <iostream>
using namespace std;

int main()
{
    int num1,num2;
    cin>>num1>>num2;
    cout<<"arithmetic operator: "<<endl;
    cout<<num1<<"+ "<<num2<<": "<<(num1+num2)<<endl;
    cout<<num1<<"- "<<num2<<": "<<(num1-num2)<<endl;
    cout<<num1<<"* "<<num2<<": "<<(num1*num2)<<endl;
    cout<<num1<<"/ "<<num2<<": "<<(num1/num2)<<endl;
    cout<<num1<<"%"<<num2<<": "<<(num1%num2)<<endl;

    cout<<"Bitwise operators: "<<endl;
    cout<<num1<<"&"<<num2<<": "<<(num1&num2)<<endl;
    cout<<num1<<"|"<<num2<<": "<<(num1|num2)<<endl;
    cout<<num1<<"^ "<<num2<<": "<<(num1^num2)<<endl;

    cout<<"conditional operator: "<<endl;
    1<2?cout<<"true"<<endl:cout<<"false"<<endl;

    cout<<"Logical Operators: "<<endl;

    if(1&&1)
    {
        cout<<"true"<<endl;
    }
    else if(1&&0 || 0&&0)
    {
        cout<<"false"<<endl;
    }
    if(!0)
    {
        cout<<"true"<<endl;
    }
    return 0;
}

```