

— — — — C PROGRAMS — — — —

1. Write a C program to get three values as an input and print the greatest and smallest number possible on the combination of the three numbers.

```
#include <stdio.h>
int main() {
    //get the inputs
    int num1, num2, num3;
    printf("Enter Number-1: ");
    scanf("%d",&num1);
    printf("Enter Number-2: ");
    scanf("%d",&num2);
    printf("Enter Number-3: ");
    scanf("%d",&num3);

    //sort the numbers
    int greatest, greater, smallest;
    if(num1>num2 && num1>num3){
        greatest = num1;
        if(num2>num3){
            greater = num2;
            smallest = num3;
        }else{
            greater = num3;
            smallest = num2;
        }
    }else if (num2>num1 && num2>num3){
        greatest = num2;
        if(num3>num1){
            greater = num3;
            smallest = num1;
        }else{
            greater = num1;
            smallest = num3;
        }
    }else{
        greatest = num3;
        if(num1>num2){
            greater = num1;
        }
    }
}
```

```

        smallest = num2;
    }else{
        greater = num2;
        smallest = num1;
    }
}

//print greatest and smallest three digit number
int greatest_number = greatest*100 + greater*10 + smallest;
int smallest_number = smallest*100 + greater*10 + greatest;

printf("Greatest three digit number : %d",greatest_number);
printf("Smallest three digit number :%d",smallest_number);
return 0;
}

```

OUTPUT :

```

Enter Number-1: 6
Enter Number-2: 3
Enter Number-3: 9
Greatest three digit number : 963
Smallest three digit number :369

```

2. Write a C Program to determine whether a given number is an automorphic number or not.

// Online C compiler to run C program online
#include <stdio.h>

```

int main() {
    //Automorphic number
    int num, product, count=0;
    printf("Enter the number:");
    scanf("%d",&num);
    product = num * num;
    int dup_num = num;
    while(dup_num>0){
        count++;
        dup_num/=10;
    }
    int divisor = 1;
    for(int i =0; i<count;i++){
        divisor *= 10;
    }
}

```

```

int new_num = product % divisor ;
if(num==new_num){
    printf("It is an automorphic number !");
}else{
    printf("It is not an automorphic number....");
}
return 0;
}

```

OUTPUT :

Enter the number: 76
It is an automorphic number !

3. Write a C Program to find the ratio of positive, negative and zero in an array

```

#include <stdio.h>

int main() {
    //Given an array with positive, negative and zero. Find the ratio of them
    int arr[8] = {1,-2,4,-5,0,6,0,5};
    int pos_count = 0;
    int neg_count = 0;
    int zero_count = 0;
    int elements = sizeof(arr)/sizeof(arr[0]);

    for(int i =0; i< elements;i++){
        if(arr[i]>0){
            pos_count++;
        }else if(arr[i]<0){
            neg_count++;
        }else{
            zero_count++;
        }
    }

    float pos_ratio = (float)pos_count / elements;
    float neg_ratio = (float)neg_count / elements;
    float zero_ratio = (float)zero_count / elements;

    printf("The ratio of positive numbers is %f",pos_ratio);
    printf("The ratio of negative numbers is %f",neg_ratio);
    printf("The ratio of zero is %f",zero_ratio);
}

```

```
    return 0;
}
```

OUTPUT :

The ratio of positive numbers is 0.500000
The ratio of negative numbers is 0.250000
The ratio of zero is 0.250000

4. Write a C program to find the Smallest and greatest element of an array

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

```
int main() {
    //Given an array find the smallest and greatest number of it.
    /*
        test cases:
        * Positive numbers
        * Negative numbers
        * Zeros
        * Empty List
        * Single element list
    */
    int arr[1]={1};
    int elements = sizeof(arr)/sizeof(arr[1]);
    if(elements==0){
        printf("The size of the array should be non-zero");
        return 0;
    }
    int greatest = arr[0];
    int smallest = arr[0];
    for(int i=1; i<elements; i++){
        if(greatest<arr[i]){
            greatest = arr[i];
        }
        if(smallest>arr[i]){
            smallest = arr[i];
        }
    }
    if(greatest == smallest){
        printf("The Greatest and Smallest number is %d",greatest);
    }else{
        printf("The Greatest number is %d",greatest);
    }
}
```

```

printf("\nThe Smallest number is %d",smallest); }

return 0;
}

```

OUTPUT :

The Greatest and Smallest number is 1

5. Write a C Program to find the exponent of a number without using pow function

```

#include <stdio.h>

int main() {
    int base, exponent, product=1;
    printf("Enter the base value:");
    scanf("%d",&base);
    printf("Enter the exponent value:");
    scanf("%d",&exponent);

    for (int i=1; i<exponent+1; i++){
        product*=base;
    }
    printf("The value of %d to the power of %d is %d",base,exponent,product);
    return 0;
}

```

OUTPUT :

```

Enter the base value:2
Enter the exponent value:10
The value of 2 to the power of 10 is 1024

```

6. Write a C program to check if a given number is a strong number or not. A strong number is a number in which the sum of the factorial of the digits is equal to the number itself.

```

#include <stdio.h>
int fact(int num){
    int product=1;

```

```

    for(int i=1;i<num+1;i++){
        product*=i;
    }
    return product;
}
int main() {
    //program to check if the number is a strong number or not
    int num,org_num;
    printf("Enter the number:");
    scanf("%d",&num);
    org_num = num;

    int sum = 0;
    while(num>0){
        int rem = num%10;
        int factorial = fact(rem);
        sum+=factorial;
        num/=10;
    }
    if(org_num==sum){
        printf("It is a Strong Number");
    }else{
        printf("It is not a Strong Number");
    }
    return 0;
}

```

OUTPUT :

Enter the number:145
It is a Strong Number

7. Write a C program to reverse a number

```

#include <stdio.h>
#include <stdbool.h>
int main(){

    int num;
    printf("Enter the number :");
    scanf("%d",&num);

```

// reverse the number

```
int res = 0;
bool neg;
if(num<0){
    num = 0-num;
    neg = true;
}
while(num>0){
    int rem = num % 10;
    res = res*10 + rem;
    num/=10;
}

if(!neg){
    printf("The reverse of the number is %d", res);
}else{
    printf("The reverse of the number is -%d", res);
}
return 0;
}
```