Click here to see solutions in github

01 Positive, Negative or Zero

Solution-1

```c
#include<stdio.h>
int main(){
	int n;
	scanf("%d", &n);
	if(n > 0){
		printf("Positive\n");
	}
	else if(n < 0){
		printf("Negative\n");
	}
	else{
		printf("Zero\n");
	}
	return 0;
}
```

02. Maximum of Three Numbers

solution-1

```c
#include<stdio.h>
int main(){
	int a, b, c;
	scanf("%d %d %d", &a, &b, &c);
	int max;
	if(a > b && a > c){
		max = a;
	}
	else if(b > a && b > c){
		max = b;
	}
	else{
		max = c;
```

```
	}
	printf("%d\n", max);
	return 0;
}
```

Solution-2

```
#include<stdio.h>
int main(){
	int a, b, c;
	scanf("%d %d %d", &a, &b, &c);
	int max_of_first_two = a > b ? a : b;
	int max = max_of_first_two  > c ?  max_of_first_two : c;
	printf("%d\n", max);
	return 0;
}
```

Solution-3

```
#include<stdio.h>
int max(int x, int y){
	return x > y ? x : y;
}
int main(){
	int a, b, c;
	scanf("%d %d %d", &a, &b, &c);
	printf("%d\n", max(max(a,b),c));
	return 0;
}
```

03. Even Numbers

Solution-1

```
#include<stdio.h>
int main(){
	for(int i = 1; i <= 50; ++i){
		if(i % 2 == 0){
			printf("%d ", i);
		}
	}
```

```
	return 0;
}
```

Solution-2

```
#include<stdio.h>
int main(){
	for(int i = 2; i <= 50; i += 2){
		printf("%d ", i);
	}
	return 0;
}
```

04. Factorial Calculation

Solution-1

```
#include<stdio.h>
int main(){
	int n;
	scanf("%d", &n);
	if(n > 0){
		int factorial = 1;
		for(int i = 1; i <= n; ++i){
			factorial *= i;
		}
		printf("%d\n", factorial);
	}
	else{
		printf("Invalid input\n");
	}

	return 0;
}
```

Solution-2

```
#include<stdio.h>
int factorial(int n){
	int fact = 1;
```

```
      for(int i = 1; i <= n; ++i){
            fact *= i;
      }
      return fact;
}
int main(){
      int n;
      scanf("%d", &n);
      if(n > 0){
            printf("%d\n", factorial(n));
      }
      else{
            printf("Invalid input\n");
      }

      return 0;
}
```

Solution-3

```
#include<stdio.h>
int factorial(int n){
      int fact = 1;
      for(int i = 1; i <= n; ++i){
            fact *= i;
      }
      return fact;
}
int main(){
      int n;
      scanf("%d", &n);
      if(n < 0){
            printf("Invalid input\n");
            return 0;
      }
      printf("%d\n", factorial(n));
      return 0;
}
```

Solution-4

```
#include<stdio.h>
int factorial(int n){
      if(n == 0){
```

```
            return 1;
      }
      return n * factorial(n - 1);
}
int main(){
      int n;
      scanf("%d", &n);
      if(n < 0){
            printf("Invalid input\n");
            return 0;
      }
      printf("%d\n", factorial(n));
      return 0;
}
```

Solution-5

```
#include<stdio.h>
#define ull unsigned long long
ull factorial(int n){
      if(n == 0){
            return 1;
      }
      return n * factorial(n - 1);
}
int main(){
      int n;
      scanf("%d", &n);
      if(n < 0){
            printf("Invalid input\n");
            return 0;
      }
      printf("%llu\n", factorial(n));
      return 0;
}
```

05 Fibonacci Series

Solution-1

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

```c
int main(){
	int n;
	scanf("%d", &n);
	printf("0 ");
	int left = 0, right = 1, current = 1;
	while(current <= n){
		printf("%d ", current);
		current = left + right;
		left = right;
		right = current;
	}
	return 0;
}
```

Solution-2

```c
#include<stdio.h>
int main(){
	int n;
	scanf("%d", &n);
	const int maxn = 1e5; // 1 * 10^5;
	int arr[maxn];
	arr[0] = 0;
	arr[1] = 1;
	for(int i = 2; i < maxn; ++i){
		arr[i] = arr[i -1] + arr[i - 2];
	}
	for(int i = 0; arr[i] <= n; ++i){
		printf("%d ", arr[i]);
	}
	return 0;
}
```

06. Leap Year

Solution-1

```c
#include<stdio.h>
int main(){
	int n;
	scanf("%d", &n);
	if(n % 400 == 0){
```

```
            printf("Leap Year\n");
      }
      else{
            if(n % 100 == 0){
                  printf("Not a Leap Year\n");
            }
            else if(n % 4 == 0){
                  printf("Leap Year\n");
            }
            else{
                  printf("Not a Leap Year\n");
            }
      }
}
```

Solution-2

```
#include<stdio.h>
int main(){
      int n;
      scanf("%d", &n);
      if((n % 400 == 0) || (n % 100 != 0 && n % 4 == 0)){
            printf("Leap Year\n");
      }
      else{
            printf("Not a Leap Year\n");
      }

}
```

07 Sum of Digits

Solution-1

```
#include<stdio.h>
int main(){
      int n;
      scanf("%d", &n);
      int sum = 0;
      while(n > 0){
            sum += n % 10;
            n /= 10;
```

```
	}
	printf("%d\n", sum);
	return 0;
}
```

Solution-2

```
#include<stdio.h>
int main(){
	char str[12];
	scanf("%s", str);
	int sum = 0;
	for(int i = 0; str[i] != '\0'/* or i < strlen(str);*/; ++i){ //
		sum += str[i] - '0';
	}
	printf("%d\n", sum);
}
```

08 Number Reversal

Solution-1

```
#include<stdio.h>
int main(){
	int n;
	scanf("%d", &n);
	int reverseNum = 0;
	while(n > 0){
		reverseNum = (reverseNum * 10) + n % 10;
		n /= 10;
	}
	printf("%d\n", reverseNum);
	return 0;
}
```

Solution-2

```
#include<stdio.h>
int main(){
	int n;
	scanf("%d", &n);
```

```
	int reverseNum = 0;
	while(n > 0){
		reverseNum = (reverseNum * 10) + n % 10;
		n /= 10;
	}
	printf("%d\n", reverseNum);
	return 0;
}
```

09 Palindrome Check

Solution-1

```
#include<stdio.h>
int main(){
	int n;
	scanf("%d", &n);
	int actual_n = n;
	int reverseNum = 0;
	while(n > 0){
		reverseNum = (reverseNum * 10) + n % 10;
		n /= 10;
	}
	if(reverseNum == actual_n){
		printf("Palindrome\n");
	}
	else{
		printf("Not a Palindrome\n");
	}
	return 0;
}
```

Solution-2

```
#include<stdio.h>
#include<string.h>
bool isPalindrome(char* str){
	int len = strlen(str);
	for(int i = 0, j = len -1; i < len; ++i, --j){
		if(str[i] != str[j]){
			return false;
		}
```

```
	}
	return true;
}
int main(){
	char str[12];
	scanf("%s", str);
	printf("%sPalindrome\n", isPalindrome(str) ? "" : "Not a ");
}
```

10 Multiplication Table

Solution-1

```
#include<stdio.h>
int main(){
	int n;
	scanf("%d", &n);
	for(int i = 1; i <= 10; ++i){
		printf("%lld\n", (long long)n*i);
	}
	return 0;
}
```

Solution-2

```
#include<stdio.h>
void printMultable(int n){
	for(int i = 1; i <= 10; ++i){
		printf("%lld\n", (long long)n*i);
	}
}
int main(){
	int n;
	scanf("%d", &n);
	printMultable(n);
	return 0;
}
```

11 Prime Number

Solution-1

```c
#include<stdio.h>
int main(){
	int n;
	scanf("%d", &n);
	bool prime = true;
	for(int i = 2; i < n; ++i){
		if(n % i == 0){
			prime = false;
			break;
		}
	}
	printf("%sPrime", prime && n > 1 ? "" : "Not a");
	return 0;
}
```

Solution-2

```c
#include<stdio.h>
bool isPrime(int n){
	if(n <= 1){
		return false;
	}
	for(int i = 2; i*i <= n; ++i){
		if( n % i == 0){
			return false;
		}
	}
	return true;
}
int main(){
	int n;
	scanf("%d", &n);
	printf("%sPrime", isPrime(n) ? "" : "Not a");
	return 0;
}
```

12 Proper Divisors

Solution-1

```c
#include<stdio.h>
int main(){
	int n;
	scanf("%d", &n);
	int number_of_divisors = 0;
	for(int i = 1; i < n; ++i){
		if(n % i == 0){
			++number_of_divisors;
		}
	}
	printf("%d\n", number_of_divisors);
	for(int i = 1; i < n; ++i){
		if(n % i == 0){
			printf("%d ", i);
		}
	}
	return 0;
}
```

Solution-2

```c
#include<stdio.h>
int main(){
	int n;
	scanf("%d", &n);
	int divisors[10000];
	int num_of_divs = 0;
	for(int i = 1; i < n; ++i){
		if(n % i == 0){
			divisors[num_of_divs] = i;
			++num_of_divs;
		}
	}
	printf("%d\n", num_of_divs);
	for(int i = 0; i < num_of_divs; ++i){
		printf("%d ",divisors[i]);
	}
	return 0;
}
```

Solution-3

```
#include<stdio.h>
#include<algorithm>
int main(){
	int n;
	scanf("%d", &n);
	int divisors[10000];
	int num_of_divs = 1;
	divisors[0] = 1;
	for(int i = 2; i*i <= n; ++i){
		if(n % i == 0){
			divisors[num_of_divs++] = i;
		}
		if(i*i != n){
			divisors[num_of_divs++] = n/i;
		}
	}
	std::sort(divisors, divisors + num_of_divs);
	printf("%d\n", num_of_divs);
	for(int i = 0; i < num_of_divs; ++i){
		printf("%d ",divisors[i]);
	}
	return 0;
}
```