/\*

Q1) Print first n natural numbers using while loop.

\*/

#include <stdio.h>

int main() {

int n, i = 1;

printf("Enter n: ");

scanf("%d", &n);

while(i <= n) {

printf("%d ", i);

i++;

}

printf("\n");

return 0;

}

/\*

Q2) Print the multiplication table of any entered number.

\*/

#include <stdio.h>

int main() {

int n, i = 1;

printf("Enter a number: ");

scanf("%d", &n);

while(i <= 10) {

printf("%d x %d = %d\n", n, i, n\*i);

i++;

}

return 0;

}

/\*

Q3) Find sum of digits of an entered number using while loop.

\*/

#include <stdio.h>

int main() {

int n, sum = 0;

printf("Enter a number: ");

scanf("%d", &n);

while(n != 0) {

sum += n % 10;

n /= 10;

}

printf("Sum of digits: %d\n", sum);

return 0;

}

/\*

Q4) Print sum of even numbers and sum of odd numbers in a given range.

\*/

#include <stdio.h>

int main() {

int start, end, sumEven = 0, sumOdd = 0, i;

printf("Enter start and end of range: ");

scanf("%d %d", &start, &end);

i = start;

while(i <= end) {

(i % 2 == 0) ? (sumEven += i) : (sumOdd += i);

i++;

}

printf("Sum of even numbers: %d\nSum of odd numbers: %d\n", sumEven, sumOdd);

return 0;

}

/\*

Q5) Print numbers divisible by 3 but not divisible by 7 in the range 1 to 500.

\*/

#include <stdio.h>

int main() {

int i = 1;

while(i <= 500) {

if(i % 3 == 0 && i % 7 != 0)

printf("%d ", i);

i++;

}

printf("\n");

return 0;

}

/\*

/\*

Q7) Find the sum of all user entered numbers until the sum exceeds 100 using do-while loop.

\*/

#include <stdio.h>

int main() {

int num, sum = 0;

do {

printf("Enter a number: ");

scanf("%d", &num);

sum += num;

} while(sum <= 100);

printf("Sum exceeded 100: %d\n", sum);

return 0;

}

/\*

Q8) Find the factorial of any user entered number using for loop.

\*/

#include <stdio.h>

int main() {

int n, i;

long long fact = 1;

printf("Enter a number: ");

scanf("%d", &n);

for(i = 1; i <= n; i++)

fact \*= i;

printf("Factorial: %lld\n", fact);

return 0;

}

/\*

Q9) Find the GCD of two entered numbers.

\*/

#include <stdio.h>

int main() {

int a, b, gcd, i;

printf("Enter two numbers: ");

scanf("%d %d", &a, &b);

for(i = 1; i <= (a<b ? a : b); i++)

if(a % i == 0 && b % i == 0)

gcd = i;

printf("GCD: %d\n", gcd);

return 0;

}

/\*

Q11) Find sum of all even numbers in a given range.

\*/

#include <stdio.h>

int main() {

int start, end, sum = 0;

printf("Enter start and end of range: ");

scanf("%d %d", &start, &end);

for(int i = start; i <= end; i++)

if(i % 2 == 0) sum += i;

printf("Sum of even numbers: %d\n", sum);

return 0;

}

/\*

Q12) Check whether an entered number is a prime or composite number.

\*/

#include <stdio.h>

int main() {

int n, flag = 0;

printf("Enter a number: ");

scanf("%d", &n);

if(n < 2) printf("Neither prime nor composite\n");

else {

for(int i = 2; i\*i <= n; i++)

if(n % i == 0) { flag = 1; break; }

printf("%d is %s\n", n, flag ? "Composite" : "Prime");

}

return 0;

}

/\*

Q13) Check whether an entered number is a palindrome number or not.

\*/

#include <stdio.h>

int main() {

int n, original, rev = 0, digit;

printf("Enter a number: ");

scanf("%d", &n);

original = n;

while(n != 0) {

digit = n % 10;

rev = rev\*10 + digit;

n /= 10;

}

printf("Number is %s palindrome\n", (rev == original) ? "" : "not");

return 0;

}

/\*

Q14) Check whether an entered number is an Armstrong number or not.

\*/

#include <stdio.h>

#include <math.h>

int main() {

int n, original, sum = 0, digit, count = 0;

printf("Enter a number: ");

scanf("%d", &n);

original = n;

int temp = n;

while(temp) { temp/=10; count++; }

temp = n;

while(temp) {

digit = temp % 10;

sum += pow(digit, count);

temp /= 10;

}

printf("%d is %sArmstrong\n", original, (sum == original) ? "" : "not ");

return 0;

}

/\*

Q15) Print all natural numbers in descending order up to 1 from an entered number except numbers divisible by 7.

\*/

#include <stdio.h>

int main() {

int n;

printf("Enter a number: ");

scanf("%d", &n);

for(int i = n; i >= 1; i--) {

if(i % 7 == 0) continue;

printf("%d ", i);

}

printf("\n");

return 0;

}

/\*

Q16) Display the binary equivalent of an entered decimal number.

\*/

#include <stdio.h>

int main() {

int n;

printf("Enter a decimal number: ");

scanf("%d", &n);

int bin[32], i = 0;

if(n == 0) printf("0");

else {

while(n > 0) {

bin[i++] = n % 2;

n /= 2;

}

for(int j = i-1; j >= 0; j--)

printf("%d", bin[j]);

}

printf("\n");

return 0;

}

/\*

Q17) Display the decimal equivalent of an entered binary number.

\*/

#include <stdio.h>

#include <math.h>

int main() {

long long n;

printf("Enter a binary number: ");

scanf("%lld", &n);

int dec = 0, i = 0, digit;

while(n > 0) {

digit = n % 10;

dec += digit \* pow(2, i++);

n /= 10;

}

printf("Decimal: %d\n", dec);

return 0;

}

/\*

Q18) Count number of digits of a number.

\*/

#include <stdio.h>

int main() {

int n, count = 0;

printf("Enter a number: ");

scanf("%d", &n);

do {

n /= 10;

count++;

} while(n != 0);

printf("Number of digits: %d\n", count);

return 0;

}

/\*

Q19) Compute sum of digits of a number.

\*/

#include <stdio.h>

int main() {

int n, sum = 0;

printf("Enter a number: ");

scanf("%d", &n);

while(n != 0) {

sum += n % 10;

n /= 10;

}

printf("Sum of digits: %d\n", sum);

return 0;

}

/\*

Q20) Compute reverse of a number and check whether the number is palindrome or not.

\*/

#include <stdio.h>

int main() {

int n, rev = 0, original, digit;

printf("Enter a number: ");

scanf("%d", &n);

original = n;

while(n != 0) {

digit = n % 10;

rev = rev\*10 + digit;

n /= 10;

}

printf("Reversed number: %d\n", rev);

printf("Number is %s palindrome\n", (rev == original) ? "" : "not ");

return 0;

}

/\*

Q22) Hint/Explanation: An Armstrong number is a number that is the sum of its own digits

each raised to the power of the number of digits.

Example: 153 = 1^3 + 5^3 + 3^3, 1634 = 1^4 + 6^4 + 3^4 + 4^4

\*/

#include <stdio.h>

#include <math.h>

int main() {

int n, temp, remainder, digits = 0, sum = 0;

printf("Enter a number: ");

scanf("%d", &n);

temp = n;

while(temp != 0) {

digits++;

temp /= 10;

}

temp = n;

while(temp != 0) {

remainder = temp % 10;

sum += pow(remainder, digits);

temp /= 10;

}

if(sum == n)

printf("%d is an Armstrong number.\n", n);

else

printf("%d is NOT an Armstrong number.\n", n);

return 0;

}

/\*

Q23) Check whether a number is a Perfect number or not.

A Perfect number is equal to the sum of its proper divisors (excluding itself).

Example: 6 = 1 + 2 + 3

\*/

#include <stdio.h>

int main() {

int n, sum = 0;

printf("Enter a number: ");

scanf("%d", &n);

for(int i = 1; i <= n/2; i++) {

if(n % i == 0)

sum += i;

}

if(sum == n)

printf("%d is a Perfect number.\n", n);

else

printf("%d is NOT a Perfect number.\n", n);

return 0;

}

/\*

Q24) Display all factors of a number.

\*/

#include <stdio.h>

int main() {

int n;

printf("Enter a number: ");

scanf("%d", &n);

printf("Factors of %d: ", n);

for(int i = 1; i <= n; i++) {

if(n % i == 0)

printf("%d ", i);

}

printf("\n");

return 0;

}

/\*

Q25) Print multiplication table of a number.

\*/

#include <stdio.h>

int main() {

int n;

printf("Enter a number: ");

scanf("%d", &n);

printf("Multiplication table of %d:\n", n);

for(int i = 1; i <= 10; i++)

printf("%d x %d = %d\n", n, i, n\*i);

return 0;

}

/\*

Q26) Check whether a number is a perfect square or not.

\*/

#include <stdio.h>

#include <math.h>

int main() {

int n;

printf("Enter a number: ");

scanf("%d", &n);

int sqrt\_n = sqrt(n);

if(sqrt\_n \* sqrt\_n == n)

printf("%d is a perfect square.\n", n);

else

printf("%d is NOT a perfect square.\n", n);

return 0;

}

27.a

// a. 1 + 1/2 + 1/3 + ... + 1/n

#include <stdio.h>

int main() {

int n, i;

float sum=0;

scanf("%d",&n);

for(i=1;i<=n;i++) sum += 1.0/i;

printf("%f",sum);

return 0;

}

27.b

// b. 1 + 2 + ... + n

#include <stdio.h>

int main() {

int n,i,sum=0;

scanf("%d",&n);

for(i=1;i<=n;i++) sum += i;

printf("%d",sum);

return 0;

}

27.c

// c. 1 + 1/2^2 + 1/3^2 + ... + 1/n^2

#include <stdio.h>

int main() {

int n,i;

float sum=0;

scanf("%d",&n);

for(i=1;i<=n;i++) sum += 1.0/(i\*i);

printf("%f",sum);

return 0;

}

27.d

// d. 1 + 2^2 + 3^2 + ... + n^2

#include <stdio.h>

int main() {

int n,i,sum=0;

scanf("%d",&n);

for(i=1;i<=n;i++) sum += i\*i;

printf("%d",sum);

return 0;

}

27.e

// e. 1 + 2^3 + 3^3 + ... + n^3

#include <stdio.h>

int main() {

int n,i,sum=0;

scanf("%d",&n);

for(i=1;i<=n;i++) sum += i\*i\*i;

printf("%d",sum);

return 0;

}

27.f

// f. 1/2 + 2/3 + ... + n/(n+1)

#include <stdio.h>

int main() {

int n,i;

float sum=0;

scanf("%d",&n);

for(i=1;i<=n;i++) sum += (float)i/(i+1);

printf("%f",sum);

return 0;

}

27.g

// g. -x + x^2 - x^3 + ... + (-1)^n \* x^n

#include <stdio.h>

#include <math.h>

int main() {

int n,i,x;

int sum=0;

scanf("%d %d",&x,&n);

for(i=1;i<=n;i++) sum += pow(-1,i)\*pow(x,i);

printf("%d",sum);

return 0;

}

27.h

// h. 1 + x + x^2/2! + ... + x^n/n!

#include <stdio.h>

int main() {

int n,i,x;

double sum=1,fact=1,powx=1;

scanf("%d %d",&x,&n);

for(i=1;i<=n;i++) {

powx \*= x;

fact \*= i;

sum += powx/fact;

}

printf("%lf",sum);

return 0;

}

27.i

// i. 1 + 1/2! + 1/3! + ... + 1/n!

#include <stdio.h>

int main() {

int n,i;

double sum=1,fact=1;

scanf("%d",&n);

for(i=2;i<=n;i++) {

fact \*= i;

sum += 1.0/fact;

}

printf("%lf",sum);

return 0;

}

27.j

// j. Sin(x) = x - x^3/3! + x^5/5! - x^7/7! + ...

#include <stdio.h>

#include <math.h>

int main() {

int n,i,sign=1;

double x,sum=0,term,fact;

scanf("%lf %d",&x,&n);

for(i=1;i<=n;i+=2) {

fact=1;

for(int j=1;j<=i;j++) fact\*=j;

term = pow(x,i)/fact;

sum += sign\*term;

sign\*=-1;

}

printf("%lf",sum);

return 0;

}

/\* 28. Print the following pattern:

1 2 3 4 5

1 2 3 4

1 2 3

1 2

1 \*/

#include <stdio.h>

int main() {

int i,j,n=5;

for(i=n;i>=1;i--) {

for(j=1;j<=i;j++) printf("%d ",j);

printf("\n");

}

return 0;

}

/\* 29. Print the following pattern:

A

A B

A B C

A B C D

A B C D E

A B C D E F \*/

#include <stdio.h>

int main() {

int i,j,n=6;

for(i=1;i<=n;i++) {

for(j=0;j<i;j++) printf("%c ", 'A'+j);

printf("\n");

}

return 0;

}

/\* 30. Print the following pattern:

1

1 2 3

1 2 3 4 5

1 2 3 4 5 6 7 \*/

#include <stdio.h>

int main() {

int i,j,n=4;

for(i=1;i<=n;i++) {

for(j=1;j<=2\*i-1;j++) printf("%d ",j);

printf("\n");

}

return 0;

}

/\* 31. Print the following pattern:

1

0 0

1 1 1

0 0 0 0

1 1 1 1 1 \*/

#include <stdio.h>

int main() {

int i,j,n=5;

for(i=1;i<=n;i++) {

for(j=1;j<=i;j++) printf("%d ",i%2);

printf("\n");

}

return 0;

}

/\* 32. Print the following pattern:

1

1 2 1

1 2 3 2 1 \*/

#include <stdio.h>

int main() {

int i,j,n=3;

for(i=1;i<=n;i++) {

for(j=1;j<=i;j++) printf("%d ",j);

for(j=i-1;j>=1;j--) printf("%d ",j);

printf("\n");

}

return 0;

}

/\* 33. Print the following pattern:

\*\*\*\*\*

\*\*\*

\* \*/

#include <stdio.h>

int main() {

int i,j,n=5;

for(i=n;i>=1;i-=2) {

for(j=1;j<=i;j++) printf("\*");

printf("\n");

}

return 0;

}

/\* 34. Print the following pattern:

\*

\* \*

\* \* \*

\* \* \* \* \*/

#include <stdio.h>

int main() {

int i,j,n=4;

for(i=1;i<=n;i++) {

for(j=1;j<=i;j++) printf("\* ");

printf("\n");

}

return 0;

}

/\* 35. Print the following pattern:

\*

\* \*

\* \* \*

\* \* \* \* \*/

#include <stdio.h>

int main() {

int i,j,space,n=4;

for(i=1;i<=n;i++) {

for(space=1;space<=n-i;space++) printf(" ");

for(j=1;j<=i;j++) printf("\* ");

printf("\n");

}

return 0;

}

/\* 36. Print the following pattern:

1

1 2

1 2 3

1 2 3 4 \*/

#include <stdio.h>

int main() {

int i,j,n=4;

for(i=1;i<=n;i++) {

for(j=1;j<=i;j++) printf("%d ",j);

printf("\n");

}

return 0;

}

/\* 37. Print the following pattern:

1

1 2 1

1 2 3 2 1

1 2 3 4 3 2 1 \*/

#include <stdio.h>

int main() {

int i,j,space,n=4;

for(i=1;i<=n;i++) {

for(space=1;space<=n-i;space++) printf(" ");

for(j=1;j<=i;j++) printf("%d ",j);

for(j=i-1;j>=1;j--) printf("%d ",j);

printf("\n");

}

return 0;

}

/\* 38. Print the following pattern:

\*

\* \*

\* \* \*

\* \* \* \*

\* \* \* \* \* \*/

#include <stdio.h>

int main() {

int i,j,space,n=5;

for(i=1;i<=n;i++) {

for(space=1;space<=n-i;space++) printf(" ");

for(j=1;j<=i;j++) printf("\* ");

printf("\n");

}

return 0;

}

/\* 39. Print all prime numbers in a given range \*/

#include <stdio.h>

int main() {

int i,j,n1,n2,prime;

scanf("%d %d",&n1,&n2);

for(i=n1;i<=n2;i++) {

if(i<2) continue;

prime=1;

for(j=2;j\*j<=i;j++) {

if(i%j==0) { prime=0; break; }

}

if(prime) printf("%d ",i);

}

return 0;

}

/\* 40. Print all palindrome numbers in a given range \*/

#include <stdio.h>

int main() {

int i,n1,n2,temp,rev,num;

scanf("%d %d",&n1,&n2);

for(i=n1;i<=n2;i++) {

temp=i; rev=0;

while(temp>0) {

rev=rev\*10+temp%10;

temp/=10;

}

if(rev==i) printf("%d ",i);

}

return 0;

}

/\* 41. Print all Armstrong numbers in a given range \*/

#include <stdio.h>

#include <math.h>

int main() {

int i,n1,n2,temp,digits,sum,num;

scanf("%d %d",&n1,&n2);

for(i=n1;i<=n2;i++) {

temp=i; sum=0; num=i;

digits=0;

while(temp>0){ digits++; temp/=10; }

temp=i;

while(temp>0) {

sum+=pow(temp%10,digits);

temp/=10;

}

if(sum==num) printf("%d ",num);

}

return 0;

}

/\* 42. Print all numbers in a range that are divisible by 5 and 7 \*/

#include <stdio.h>

int main() {

int i,n1,n2;

scanf("%d %d",&n1,&n2);

for(i=n1;i<=n2;i++) {

if(i%5==0 && i%7==0) printf("%d ",i);

}

return 0;

}

/\* 44. Print multiplication table in the given range \*/

#include <stdio.h>

int main() {

int i,j,n1,n2;

scanf("%d %d",&n1,&n2);

for(i=n1;i<=n2;i++) {

for(j=1;j<=10;j++) {

printf("%d x %d = %d\n",i,j,i\*j);

}

printf("\n");

}

return 0;

}

/\* 45. Convert decimal number to octal \*/

#include <stdio.h>

int main() {

int n;

scanf("%d",&n);

printf("%o",n);

return 0;

}

/\* 46. Convert decimal number to binary \*/

#include <stdio.h>

int main() {

int n,b[32],i=0;

scanf("%d",&n);

while(n>0) {

b[i++]=n%2;

n/=2;

}

for(i=i-1;i>=0;i--) printf("%d",b[i]);

return 0;

}