# Q1. WAP to print all even numbers using one loop and all odd numbers using another loop (both in the same program). Take max number as input.

#include <stdio.h>

#include <stdlib.h>

int main()

{

int max;

//printf("Enter the maximum limit : ");

scanf("%d",&max);

for(int i = 2; i<=max; i+=2)

printf("%d ",i);

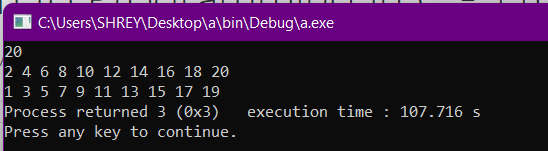
printf("\n");

for(int j = 1; j<=max; j+=2)

printf("%d ",j);

return 3;

}



# Q2. WAP to check if the number entered by a user is even or odd. Continue checking till user wishes to exit.

#include <stdio.h>

#include <stdlib.h>

int main()

{

int num;

char user;

while(1)

{

printf("Enter the number : ");

scanf("%d",&num);

if(!(num%2))

{

printf("Even\n");

printf("Enter Y to continue & N to exit : ");

scanf(" %c",&user);

if(user=='Y')

continue;

else if(user=='N')

break;

}

else

{

printf("Odd\n");

printf("Enter Y to continue & N to exit : ");

scanf(" %c",&user);

if(user=='Y')

continue;

else if(user=='N')

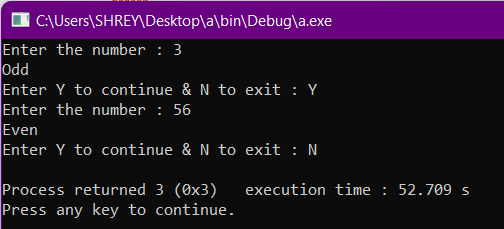
break;

}

}

return 3;

}



# Q3. WAP to print the [factorial of a number](http://164.164.122.107/moodle/mod/vpl/view.php?id=592) using do-while loop. Be careful about zero.

#include <stdio.h>

#include <stdlib.h>

int main()

{

int n, p = 1, i = 0;

//printf("Enter the number : ");

scanf("%d",&n);

if(n!=0)

{

do

{

i = i+1;

p = p\*i;

}

while(i<n);

printf("Factorial of %d is %d.",n,p);

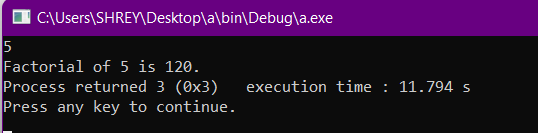
}

else

printf("Factorial of 0 is 1.");

return 3;

}



# Q4. WAP to print all prime numbers in a given range [n1 to n2].

#include <stdio.h>

#include <stdlib.h>

int main()

{

int n1, n2, i, j;

//printf("Enter the range from n1 to n2 : ");

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

if(n1==1)

n1 = 2;

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

{

for(j = 2; j <= i; ++j)

{

if(!(i%j))

break;

}

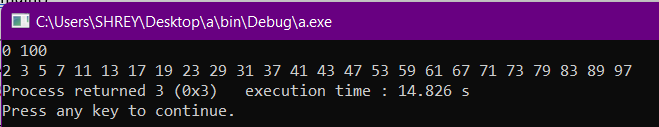
if(i==j)

printf("%d ",i);

}

return 3;

}



# Q5. WAP to find the power of a given number without using the pow() function. Use loops. If power is negative, print as a fraction instead of decimals.

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

int main()

{

int number, power, count = 0, product = 1;

//printf("Enter the number & power : ");

scanf("%d %d",&number,&power);

do

{

count = count + 1;

product = product\*number;

}

while(count<abs(power));

if(power<0)

printf("%d raised to %d is 1/%d",number,power,product);

else if(power>0)

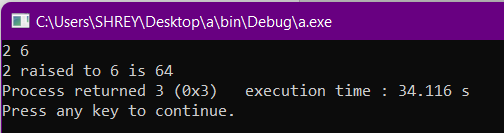
printf("%d raised to %d is %d",number,power,product);

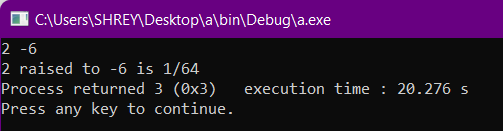
else if(power==0)

printf("%d raised to 0 is 1",number);

return 3;

}





# Q6. WAP to print if a given number is an Armstrong number. An Armstrong number is a number that is equal to the sum of the cubes of its own digits.

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

int main()

{

int num ;

//printf("Enter the number : ");

scanf("%d",&num);

//FINDING DIGIT

int digit = 0;

for(int q = num; q!=0;)

{

q = q/10;

digit = digit + 1;

}

// printf("%d",digit);

//POWER AND ADD

int r;

int y;

int sum = 0;

for(int i = num; i!=0;)

{

r = i%10;

y = pow(r,digit);

sum = sum + y;

i = i/10;

}

//CHECKING FINALLY

if(num==sum)

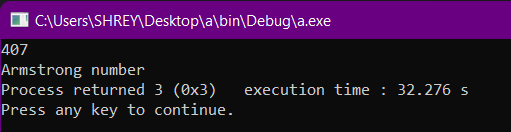
printf("Armstrong number");

else

printf("Normal number");

return 3;

}



# Q7. WAP to use loops and if-else to print Pascal's Triangle for n rows. Input is the number of rows.

#include <stdio.h>

#include <stdlib.h>

int main()

{

int n, coef=0;

//printf("Enter the number of rows=");

scanf("%d",&n);

for(int i=0; i<n; i++)

{

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

{

printf(" ");

}

for(int k=0; k<=i; k++)

{

if(k==0||k==i)

{

coef=1;

printf("%d ",coef);

}

else

{

coef = coef \* (i - k + 1) / k;

printf("%d ",coef);

}

}

printf("\n");

}

return 3;

}

