**AMCAT MOSTLY ASKED AUTOMATA QUESTIONS**

**Ques. Print the following Pattern and get the OutPut?**

N=5
**Output**
1
3*2
4*5*6
10*9*8*7
11*12*13*14*15

## PROGRAM IN C FOR AMCAT PATTERN

```
#include
int main()
{
int i,j,n,count=0,k=0;
printf("Enter N");
scanf("%d",&n);
for(i=1;i<=n;i++)
{
count=k;
for(j=1;j<=i;j++)
{
if(i%2==0)
{
printf("%d",count+i);
count=count-1;
if(j!=i)printf("*");
k++;
}
else
{
count=count+1;
printf("%d",count);
if(j!=i)printf("*");
k++;
}
}
printf("\n");
}
return 0;
}
```

**Ques. Print the following Pattern and get the output to match test cases?**

To print the pattern like
for n=3
the program should print
**1 1 1 2**
**3 2 2 2**
**3 3 3 4**

## PROGRAM IN C++

```
#include <iostream>

using namespace std;

int main()
{
   int n=3,c=n-1;
   for(int i=1;i<=n;i++)
   {
       if(i%2==0)
       cout<<c++;
       for(int j=1;j<=n;j++)
       {
           cout<<i;
       }
       if(i%2!=0)
       cout<<c++;
       cout<<"\n";
   }


   return 0;
}
```

**Ques. Programming Pattern to Print  2*N  Number of rows for input Pattern?**

```
3
44
555
6666
555
44
3
```

```
#include <iostream>

using namespace std;

int main()
{
int n=4,num=n-1;
for(int i=1;i<=n;i++)
{
for(int j=1;j<=i;j++)
cout<<num;
num++;
cout<<endl;
}
num--;
for(int i=n;i>=1;i--)
{
for(int j=1;j<=i;j++)
cout<<num;
num--;
cout<<endl;
}


return 0;

}
```

**Amcat Trapezium pattern Solution**
**To print the trapezium pattern.**
for example , we have num=4
the output should be like

```
1*2*3*4*17*18*19*20
- -5*6*7*14*15*16
- - - -8*9*12*13
- - - - - -10*11
```

```
#include<iostream>
using namespace std;
int main(){
int n=4,num=1,i=1,space=0,k=1,number=n;
for(i=0;i<n;i++)
{
for(int j=1;j<=space;j++)
{
cout<<"-";
}
```

```
for(int m=1;m<2*n-space;m++)
{
if(m%2==0)
cout<<"*";
else
cout<<num++;
}
cout<<"*";
for(int l=1;l<2*n-space;l++)
{
if(l%2==0)
cout<<"*";
else
{
cout<<k+number*number;
k++;
}
}
number--;

space=space+2;
cout<<endl;
}
return 0;
}
```

**Problem: To find GCD of two number.**

Solution:

```
#include <iostream>

using namespace std;
int gcd_iter(int u, int v)
{
int t;
while (v)
{
t = u;
u = v;
v = t % v;

}
return u < 0 ? -u : u;
}
int main()
{
```

```
int n=3,m=6;
int result=gcd_iter(n,m);
cout<<result;
return 0;
}
```

**Question: Find the number of occurrences of input num2 in input num1 and return it with function int isOccured(int num1, int num2).**

**Input:** num1 and num2

such that 0 <= num1 <= 99999999

and 0 <= num2 <= 9

**Example:**
Input: num1= 199294, num2= 9

Output: 3

**Test Case 1:**

Input:
1222212

2

Expected Output:

5

**Test Case 2:**

Input:

1001010

0

Expected Output:

4

**Its solution is like the frequency count of number .**

**Print all prime no which are below then given input no separated by comma .**

input
20

output
2,3,5,7,11,13,17,19

```
<iostream>
using namespace std;

int main() {
int n=20,count,k=0,arr[10];
for(int i=2;i<n;i++)
{count=0;
for(int j=2;j<i;j++)
{
if(i%j==0)
{
count++;
break;
}
}

if(count==0)
{
arr[k]=i;
k++;
}
}
cout<<"Prime no are \n";
for(int i=0;i<k-1;i++)
{
if(i!=k-2)
cout<<arr[i]<<",";
else
cout<<arr[i];

}

return 0;
}
```

**Program to count frequency of each element of array**

Example

Input

Input array elements: 5, 10, 2, 5, 50, 5, 10, 1, 2, 2
Output

Frequency of 5 = 3
Frequency of 10 = 2
Frequency of 2 = 3
Frequency of 50 = 1
Frequency of 1 = 1

# Program to Display Armstrong Number Between Two Intervals

```
#include <stdio.h>
int main()
{
int n1, n2, i, temp, num, rem;
printf("Enter two numbers(intervals): ");
scanf("%d %d", &n1, &n2);
printf("Armstrong numbers between %d an %d are: ", n1, n2);
for(i=n1+1; i<n2; ++i)
{
temp=i;
num=0;
while(temp!=0)
{
rem=(temp%10);
num+=rem*rem*rem;
temp/=10;
}
if(i==num)
{
printf("%d ",i);
}
}
return 0;
}
```

# Program to Display Prime Numbers between Two Intervals

```
<stdio.h>
int main()
{
int n1, n2, i, j, flag;
printf("Enter two numbers(intevals): ");
scanf("%d %d", &n1, &n2);
printf("Prime numbers between %d and %d are: ", n1, n2);
for(i=n1+1; i<n2; ++i)
{
flag=0;
for(j=2; j<=i/2; ++j)
{
if(i%j==0)
{
flag=1;
break;
}
}
if(flag==0)
printf("%d ",i);
}
return 0;
}
```

program to find out prime factors of given number

Algorithm
1. Find factor of a given number
2. Apply prime number program to check whether factor is prime or not

```
main()
{
int num,i=1,j,k;
printf("\nEnter a number:");
scanf("%d",&num);
while(i<=num)
{
k=0;
if(num%i==0) /* Finding factor */
{
j=1;
while(j<=i) /* Check prime or not */
{
if(i%j==0)
k++;
j++;
}
if(k==2)
printf("\n%d is a prime factor",i);
}
i++;
}
}
```

Write a function that accepts a sentence as a parameter, and returns the same with each of its words reversed. The returned sentence should have 1 blank space between each pair of words.
Demonstrate the usage of this function from a main program.

Example:
INPUT : “jack and jill went up a hill”
OUTPUT : “kcaj dna llij tnew pu a llih”

```
#include <iostream>
#include<string.h>
using namespace std;

void strRev(char *arr, int startIndex, int endIndex) {
for(int i=endIndex; i>=startIndex; i--)
{
cout<< arr[i];
}
cout<<" ";
}

int main() {
char st[70]="jack and jill went up hill";
int p;
p = strlen(st);
int startIndex = 0;
//int wordCount = 0;
for(int j=0;j<=p;j++)
{
if(j == p) {
strRev(st, startIndex, j -1);
// wordCount++;
```

```
}
if(st[j] == ' ') {
strRev(st, startIndex, j);
wordCount++;
startIndex = j + 1;
}
}
//cout<< "Word Count " << wordCount;
return 0;
}
```

Given an array of integers, write a function that returns true if there is a triplet (a, b, c) that satisfies a2 + b2 = c2.

Example:

Input: arr[] = {3, 1, 4, 6, 5}
Output: True
There is a Pythagorean triplet (3, 4, 5).

Input: arr[] = {10, 4, 6, 12, 5}
Output: False
There is no Pythagorean triplet.

```
#include <iostream>
using namespace std;

// Returns true if there is Pythagorean triplet in ar[0..n-1]
bool isTriplet(int ar[], int n)
{
for (int i=0; i<n; i++)
{
for (int j=i+1; j<n; j++)
{
for (int k=j+1; k<n; k++)
{
// Calculate square of array elements
int x = ar[i]*ar[i], y = ar[j]*ar[j], z = ar[k]*ar[k];

if (x == y + z || y == x + z || z == x + y)
return true;
}
}
}

// If we reach here, no triplet found
return false;
}

int main()
{
int ar[] = {3, 1, 4, 6, 5};
int ar_size = sizeof(ar)/sizeof(ar[0]);
isTriplet(ar, ar_size)? cout << "Yes": cout << "No";
return 0;
}
```

You have 3 dice , you need to take input from the user whats the sum of three dice he wants . Program to find all the possible outcome of input..

example
INPUT :

5
OUTPUT :
6
EXPLATION
(1,2,2) (2,2,1 ) (2,1,2) (1,1,3) (1,3,1) (3,1,1)

INPUT
2
OUTPUT
0
EXPLANATION
minimum sum value on the dice is 3 , so it not possible to come sum equal to 2 .

INPUT :
3
OUTPUT:
1
EXPLANATION
(1,1,1)

```
#include<iostream.h>

int main()
{

int x,count=0;
cin>>x;
for(int i=1;i<=6;i++)
{
for(int j=1;j<=6;j++)

{
for(int k=1;k<=6;k++)

{
if(x==(i+j+k))
count++;
}

}

}

cout<<count;
return 0;

}
```

---

Most of the times there would be two questions. One will be rather easy and other will be average.

Easy:

Find prime numbers up to n numbers.

Eliminate vowels from a string.

Find GCD of entered numbers.

String Reversals

Sorting Algorithms
Palindrome
Pythagorean Triplets
Armstrong Number
Average:
Various Triangular Patterns like Pascal Triangle.
Mirror image of a matrix.
To print all possible outcomes from an entered string with same words.
Combine two arrays and sort.
Dynamic Memory Allocation
Swapping in two strings
BCD and Grey Character Program
They give you options to choose from C,C++ and Java