**Level I**

**Basic Level If-else**

* Swap 2 variable without using 3rd variable .(three answer)
* Accept three no and find out the highest no.
* Accept a number from user - if it is divisible by 3 print “three” , if it is divisible by 7 print “seven” and if it is divisible by both(3,7) print “three -seven”
* Accept a number from user check if it is odd or even number do not use % operator
* **Problem:** Given the length of four sides determine whether they can be used to create a polygon and determine if that polygon is a square. A polygon can be created if no single side is greater than the sum of the other three sides. Display a ONE when the status (polygon or square) is confirmed and ZERO when the status cannot beconfirmed.

**Example Execution #1:**

Enter the length of the four sides: 4 3 2 9

Length of sides: 4, 3, 2, 9

Polygon status: 1

Square status: 0

**Example Execution #2:**

Enter the length of the four sides: 5 5 5 5

Length of sides: 5, 5, 5, 5

Polygon status: 1

Square status: 1

**Example Execution #3:**

Enter the length of the four sides: 84 25 20 15

Length of sides: 84, 25, 20, 15

Polygon status: 0

Square status: 0

**LOOP**

* Accept 10 number user and do sum of it.(do not use array)
* Accept a number from user and find a factorial of a number
* Accept a number from user and reverse it.
* Accept a number from user and check if it is palindrome number or not eg (121)
* Accept term from user and print Fibonacci series
* Accept a number from user accept a digit from user and check the occurrence of that digit
* Accept 10 number from user and print highest number
* Accept 10 number from user and print lowest number.
* Accept 10 number from user and print highest and 2nd highest number.
* Write a prog to print every integer between 1 and n divisible by m. also report whether the number that is divisible by m is even or odd.

**Nested Forloop break/continue**

* Pattern

|  |  |  |
| --- | --- | --- |
| 1  12  123 | 1  22  333 | 1  2 2  3 3 3 |
| 1  23  456 | 1  12  123 | 1  21  321 |
| 1  121  12321 | 1  121  12321  121  1 | 32123  212  3 |
| 1  222  33333 | 0  101  21012 | zyxyz  zyz  z |
| a  ab  abc | \*  \*\*  \*\*\*  \*\*\*\* | \*\*  \* \*  \*\* |
| --\*\*--  -\*--\*-  \*----\* | 1  121  12121 | 321  32  3 |
|  | 1 1  2  3 3 |  |

* Accept a number from user and check if it is prime number or not