ASSIGNMENT 3

NAME:SOURABH PATEL
ADMISSION NO: U19CS082

1) **Write a code that performs username validation for a website. When the username is too short it should throw an exception such that it prints Too short: n (where n is the length of the given username). The final program should print Valid (if the username is valid), Invalid (if the username is invalid), or Too short: n (where is the length of the too-short username). Make necessary assumptions if required.**

CODE:

```
#include <iostream>
#include <string>
#include <sstream>
#include <exception>
using namespace std;

/* Define the exception here */
struct BadLengthException : exception {
    string s;
    BadLengthException(int n) : s(to_string(n)) {}
    const char *what() const noexcept override {
        return s.c_str();
    }
};


bool checkUsername(string username) {
    bool isValid = true;
    int n = username.length();
    if(n < 5) {
        throw BadLengthException(n);
    }
    for(int i = 0; i < n; i++){
        if(username[i]==' '){
            isValid = false;
        }
    }
    return isValid;
}

int main(){
    int T;
    cout<<"enter number of test cases:";
    cin >> T;
    while(T--) {
        string username;
        cout<<"Enter UserName:";
```

```
        cin >> username;
        try {
            bool isValid = checkUsername(username);
            if(isValid)
            {
                cout << "Valid" << '\n';
            }
            else
            {
                cout << "Invalid" << '\n';
            }
        }
        catch (BadLengthException e) {
            cout << "Too short: " << e.what() << '\n';
        }
    }
    return 0;
}
```

OUTPUT:

```
C:\Users\Sourabh Patel\Desktop\assignment\82\SEM6\PPL\ASS3\Q1.exe
enter number of test cases:2
Enter UserName:jackalish
Valid
Enter UserName:jack
Too short: 4

--------------------------------
Process exited after 24.72 seconds with return value 0
Press any key to continue . . .
```

2) You are required to handle error messages while working with a small computational server that performs complex calculations. It has a function that takes 2 large numbers as its input and returns a numeric result. Unfortunately, there are various exceptions that may occur during execution. Write a program so that it prints appropriate error messages. The expected behavior is defined as follows: If the compute function runs fine with the given arguments, then print the result of the function call.

If it fails to allocate the memory that it needs, print Not enough memory. If any other standard C++ exception occurs, print Exception: S where S is the exception's error message.

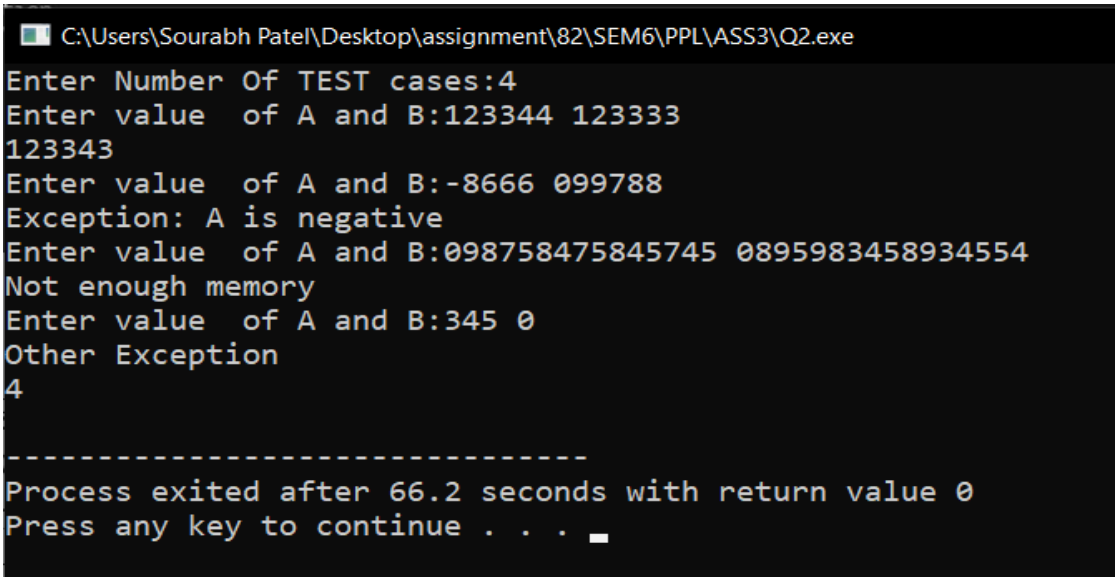
If any non-standard exception occurs, print Other Exceptions.

CODE :

```cpp
#include <iostream>
#include <exception>
#include <string>
#include <stdexcept>
#include <vector>
#include <cmath>
using namespace std;

class Server {
    private:
        static int load;
    public:
        static int compute(long long A, long long B) {
            load += 1;
            if(A < 0){
                throw std::invalid_argument("A is negative");
            }
            vector<int> v(A, 0);
            int real = -1, cmplx = sqrt(-1);
            if(B == 0) throw 0;
            real = (A/B)*real;
            int ans = v.at(B);
            return real + A - B*ans;
        }
        static int getLoad(){
            return load;
        }
};
int Server::load = 0;

int main(){
    int T;
    cout<<"Enter Number Of TEST cases:";
    cin >> T;
    while(T--) {
        long long A, B;
        cout<<"Enter value  of A and B:";
        cin >> A >> B;

        /* Enter your code here. */
        try {
            cout << Server::compute(A, B) << endl;
        }
        catch (std::bad_alloc& error) {
            cout << "Not enough memory" << endl;
        }
        catch (std::exception& error) {
            cout << "Exception: " << error.what() << endl;
```

```
        }
        catch (...) {
            cout << "Other Exception" << endl;
        }
        /* ------------*/
    }
    cout << Server::getLoad() << endl;
    return 0;
}
```

OUTPUT :



3) **Create a class Polar that represents the points on the plane as polar coordinates (radius and angles). Create an overloaded + operator for addition of two Polar quantities. "Adding" two points on the plane can be accomplished by adding their X coordinates and then adding their Y coordinates. This gives the X and Y coordinates of the "answer." Thus you'll need to convert two sets of polar coordinates to rectangular coordinates, add them, then convert the resulting rectangular representation back to polar. You need to use the following trigonometric formulae:**

x = r*cos(a);
y = r*sin(a);
a = atan(y/x); //arc tangent
r=sqrt(x*x+y*y);

CODE:

```cpp
#include<iostream>
#include<math.h>
#define pi 3.1416
using namespace std;
class polar{
                  float r,a,x,y;
                  public:
                  polar(){};
                  polar(float r1,float a1);
                  polar operator+(polar r1);
                  void display(void);
};

polar :: polar(float r1,float a1){
               r=r1;
              a=a1*(pi/180);
              x=r*cos(a);
              y=r*sin(a);
}

polar polar :: operator+(polar r1){
              polar R;

              R.x=x+r1.x;
              R.y=y+r1.y;
              R.r=sqrt(R.x*R.x+R.y*R.y);
              R.a=atan(R.y/R.x);

    return R;
}

void polar::display(){
    cout<<"radius = "<<r<<"\n angle = "<<a*(180/pi)<<"\n";
}

int main(){
  polar p1,p2,p3;
  float r,a;
  cout<<" Enter radius and angle : ";
  cin>>r>>a;
  p1=polar(r,a);
  p2=polar(8,45);
  p3=p1+p2;
  cout<<" p1 : \n";
  p1.display();
 cout<<" p2 : \n ";
  p2.display();
  cout<<" p3 : \n ";
```

```
  p3.display();
  return 0;
}
```

OUTPUT:

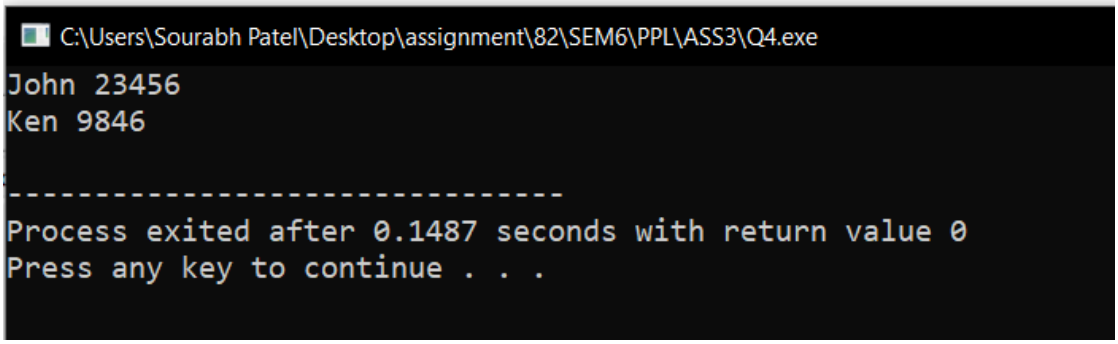
```
C:\Users\Sourabh Patel\Desktop\assignment\82\SEM6\PPL\ASS3\Q3.exe
 Enter radius and angle : 2.82843 0.5
 p1 :
radius = 2.82843
 angle = 0.5
 p2 :
 radius = 8
 angle = 45
 p3 :
 radius = 10.2117
 angle = 33.8056

--------------------------------
Process exited after 22.95 seconds with return value 0
Press any key to continue . . .
```

4. **A file contains a list of telephone numbers in the following form:**
   John 23456
   Ken 9846
   The names contain only one word and the names and telephone numbers are separated by white spaces. Write a program to read a file and display its contents in two columns. The names should be left justified and the number right justified.

CODE :

```
#include<iostream>
#include<fstream>
#include<string>
using namespace std;
int main(){
    ifstream file;
    file.open("temp.txt");
    string name,number;
    while(file >>name >>number){
         cout<<name<<" "<<number<<"\n";
    }
    file.close();
return 0;
}
```

OUTPUT:


C:\Users\Sourabh Patel\Desktop\assignment\82\SEM6\PPL\ASS3\Q4.exe
John 23456
Ken 9846

Process exited after 0.1487 seconds with return value 0
Press any key to continue . . .