Task 1:

1. Check entry of both username and password
2. Check username and password are correct
3. Check all mandatory fields are entered in delivery details form
4. Check email credentials on delivery details form
5. Check card digits are not less than 16 digits
6. Reconfirm all the details of delivery details form before final proceeding
7. Check all mandatory fields are filled in contact details form
8. Check email credentials on contact details form
9. Check if item is available to buy
10. Verify card type and card number

Task 2:

**Challenge 1:**

#include<stdio.h>

#include <math.h>

#include <stdbool.h>

int fib(int n)

{

/\* Declare an array to store Fibonacci numbers. \*/

int f[n+2]; // 1 extra to handle case, n = 0

int i;

/\* 0th and 1st number of the series are 0 and 1\*/

f[0] = 0;

printf(" %d", f[0]);

f[1] = 1;

printf(" %d", f[1]);

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

{

/\* Add the previous 2 numbers in the series

and store it \*/

f[i] = f[i-1] + f[i-2];

printf(" %d", f[i]);

}

return f[n];

}

// A utility function that returns true if x is perfect

// square

bool isPerfectSquare(int x)

{

int s = sqrt(x);

return (s \* s == x);

}

// Returns true if n is a Fibonacci Number, else false

bool isFibonacci(int n)

{

// n is Fibonacci if one of 5\*n\*n + 4 or 5\*n\*n - 4 or

// both is a perfect square

return isPerfectSquare(5 \* n \* n + 4)

|| isPerfectSquare(5 \* n \* n - 4);

}

int findIndex(int n)

{

// if Fibonacci number is less than 2,

// its index will be same as number

if (n <= 1)

return n;

int a = 0, b = 1, c = 1;

int res = 1;

// iterate until generated fibonacci number

// is less than given fibonacci number

while (c < n)

{

c = a + b;

// res keeps track of number of generated

// fibonacci number

res++;

a = b;

b = c;

}

return res;

}

int main ()

{

int n = 9;

int F = 34;

printf("\n %d ", fib(n));

getchar();

printf("\n");

if (isFibonacci(F))

printf(" %d is a Fibonacci Number \n", F);

else

printf(" %d is not Fibonacci Number \n", F);

printf(" %d is the closest index ", findIndex(F));

return 0;

}

**Challenge 2:**

/\* C Program to Find Maximum Occurring Character in a String \*/

#include <stdlib.h>

#include <stdio.h>

#include <string.h>

int main()

{

char str[20] = "Character", result;

int i;

int max = -1;

//CONVERT STRING TO LOWERCASE

for(i=0;i<=strlen(str);i++)

{

if(str[i]>=65&&str[i]<=90)

str[i]=str[i]+32;

}

int freq[256] = {0};

//printf("\n Please Enter any String : ");

//gets(str);

//COUNT FREQUENCY OF EACH CHARACTER

for(i = 0; i < strlen(str); i++)

{

freq[str[i]]++;

}

//GET MAXIMUM FREQUENCY CHARACTER INTO RESULT

for(i = 0; i < strlen(str); i++)

{

if(max < freq[str[i]])

{

max = freq[str[i]];

result = str[i];

}

}

printf("\n The maximum occurring character is %c ", result);

return 0;

}