ICS PROJECT

PASSWORD GENERATOR

- JAISWAL ADITYA RANJIT (B22CS025)
- GUBBALA BHARGAVI (B22CS022)
- HARSHIKA ARYA (B22CS023)
- HIMANSHU SHARMA (B22ME026)
- GAMIT SARABEN (B22EE027)

GROUP CODE: G20

PROJECT INSTRUCTOR : SHUBHAM

PROBLEM STATEMENT:

"Develop a C program that allows users to enter a password of any length(min 8 characters), and the program evaluates the strength of the password based on the criteria. If the strength of the password is less than 80%, the program should suggest a stronger password. The suggested password should have the same length as the input password and should meet the required strength level. The program should display both the original and suggested passwords with their respective strength levels. The program should display the generated password and its strength level(must be greater than 80% strength level).

55% strength: The password contains only lowercase alphabets.

65% strength: The password contains both lowercase and uppercase alphabets.

75% strength: The password contains alphabets (both lowercase and uppercase) along with numbers.

85% strength: The password contains alphabets (both lowercase and uppercase) along with numbers and special characters (!,@,#,$,%,^,&,*).

100% strength: The password contains alphabets (both lowercase and uppercase), numbers and special characters (!,@,#,$,%,^,&,*), each with a minimum length of 3 characters. "

EXAMPLE OF OUTPUT

```
**************************************************************************************************************************************************************
                                                  W E L C O M E   T O   P A S S W O R D   G E N E R A T O R
**************************************************************************************************************************************************************
Please follow the instructions while entering  password :
        1. Password must be minimum 8 characters long.
        2. Password should only contain alphabets(lowercase or uppercase), digits (0 to 9)
           and special symbols (!,@,#,%,^,&).
Please enter the password :
adi123ya4
Strength of the password you have entered is 65%
The new suggested password is M*$3kY@1u
Strength of suggested password is 85 %
if you want your current password to be accepted press 0
if you want use suggested password press 2
if you want to try a new password press 1
1
```

CODE ALGORITHM:

https://onedrive.live.com/?authkey=%21AGCrH%2D%2DbDoxRAnk&id=56F990CDF10C4C60%211125&cid=56F990CDF10C4C60&parId=root&parQt=sharedby&o=OneUp

RANDOM PASSWORD

STARTING BY AN EXAMPLE:
We know that if we are having number n as divisor it is going to generate 0,1,2....,n-1 as remainders.
We can relate this to fact that if we are having array of size 'len' the index of all elements are nothing but reminders generated by dividing any number with integer len.
If dividend is any random number and we are increasing it (len -1) times and keep on dividing the numbers by len, This is going to yield remainders from 0 to len-1. hence by this way we can access the index of final array randomly.

```
void random(int len)
{
    char a[]="ABCDEFGHIJKLMNOPQRSTUVWXYZ";
    char b[]="abcdefghijklmnopqrstuvwxyz";
    char c[]="0123456789";
    char d[]="!@#$%^&*";
    char fi[len+1];
    fi[len]='\0';
    int p=l2;

    for(int j=l1;j<l1+len;j++)
    {
        l2+=j;
        p=p+3;

        if(p%4==0)
        fi[j%len]=a[l2%26];
        else if(p%4==1)
        fi[j%len]=b[l2%26];
        else if(p%4==2)
        fi[j%len]=c[l2%10];
        else
        fi[j%len]=d[l2%8];
    }
    printf("The new suggested password is %s\n",fi);
    printf("Strength of suggested password is %d %c\n",str

}
```

- We have declared 4 character arrays which will store uppercase alphabets,lowercase alphabets,digits and special symbols.
- lets say we have 'len' number of boxes named from 0 to len-1.
- Now we have to randomly pick from which array are we going to pick the element we want to put in those boxes randomly.
- These boxes are also getting decided randomly by reminder logic.
- Depending on the remainders generated when that random number is divided by 4 we are going to select from which array the element will be picked.
- Also after deciding array, now which element is to be picked from the decided arrayis also decided by same remainder logic.

```
void random(int len)
{
    char a[]="ABCDEFGHIJKLMNOPQRSTUVWXYZ";
    char b[]="abcdefghijklmnopqrstuvwxyz";
    char c[]="0123456789";
    char d[]="!@#$%^&*";
    char fi[len+1];
    fi[len]='\0';
    int p=l2;

    for(int j=l1;j<l1+len;j++)
    {
        l2+=j;
        p=p+3;

        if(p%4==0)
        fi[j%len]=a[l2%26];
        else if(p%4==1)
        fi[j%len]=b[l2%26];
        else if(p%4==2)
        fi[j%len]=c[l2%10];
        else
        fi[j%len]=d[l2%8];
    }
    printf("The new suggested password is %s\n",fi);
    printf("Strength of suggested password is %d %c\n"

}
```

THANK YOU!