PF LAB 8 Assignment

Name: Muhammad Danish
Roll no. 22K-4381
Section- 1D

Task 01:

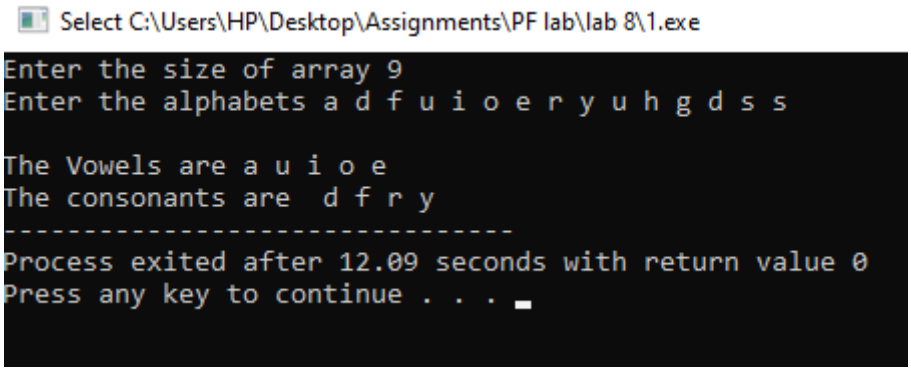
Create an array named alpha that is of char data type. User can input random characters into this array. Create a program that tells the user how many vowels and consonants has the user emtered. Also create two new arrays named vowels and consonant and store the vowels in vowels array, store the consonant in consonant array.

Program:

```
#include <stdio.h>
int main()
{
int size;
printf("Enter the size of array");
scanf(" %d", &size);

char alpha[size];
printf("Enter the alphabets ");

char x, vowel[size], cons[size];
int vlen=0,clen=0;

for (int i=0; i<size; i++)
{
scanf(" %c", &alpha[i]);


if(alpha[i]=='a'||alpha[i]=='e'||alpha[i]=='i'||alpha[i]=='o'||alpha[i]=='u'||alpha[i]=='A'||alpha[i]=='E'||alph
a[i]=='I'||alpha[i]=='O'||alpha[i]=='U')
{
        vowel[vlen]=alpha[i];
        vlen++;
}
else
{
        cons[clen]=alpha[i];
        clen++;
}
}

int i=0;
printf("\nThe Vowels are");
```

```
while (i<vlen)
{
printf(" %c", vowel[i]);
i++;
}

i=0;
printf("\nThe consonants are ");
while (i<clen)
{
printf(" %c", cons[i]);
i++;
}
}
```

Output:



Task 02:

Write a program that takes a word from user and display it in reverse order.

Sample Output:

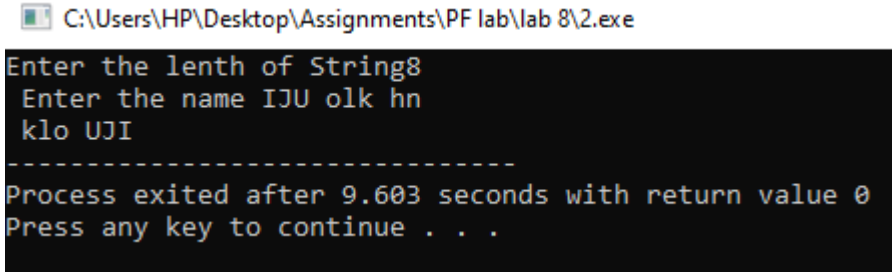
Enter name: Ali

Output ilA

Program:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
int main()
{
int size;
printf("Enter the lenth of String");
scanf("%d", &size);
 fflush(stdin);

char x[100];
printf(" Enter the name ");
gets(x);

int s=size;
char rev[size];
for (int i=0 ; i<s; i++)
{
rev[i]= x[size-i-1];
}
for (int i=0; i<s; i++)
        printf("%c", rev[i]);
}
```

Output:



Task 03:

Suppose A, B, C are arrays of integers of size M, N, and (M + N) respectively. Your program should perform the following tasks:

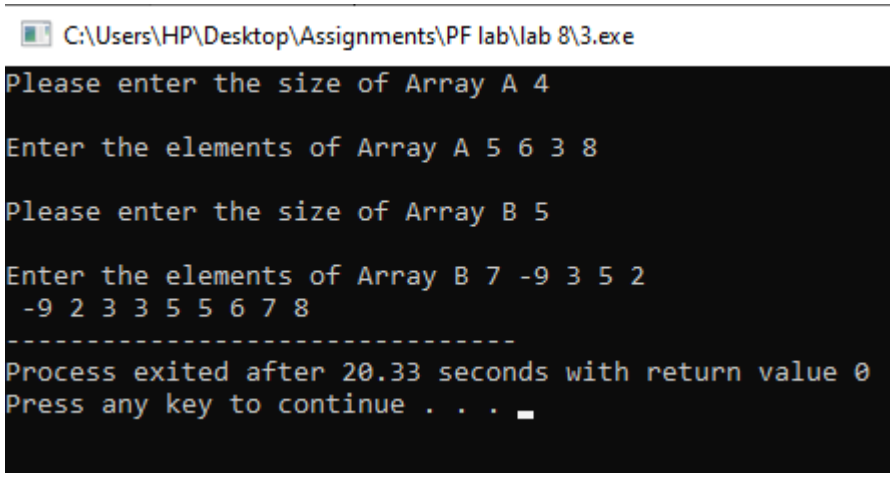
- Take number of elements and elements from user input for array A and sort it in ascending order.
- Take number of elements and elements from user input for array A and sort it in ascending order.
- In array C, merge array A and array B in ascending order.
- Display array A and array be before sorting and after sorting. Display array C after merge.

Program:

```
#include <stdio.h>
//program for ascending order of arrays
int asc(int a[], int size);

int main()
{
	int sizeA, sizeB;
	printf("Please enter the size of Array A ");
	scanf(" %d", &sizeA);
	int a[sizeA];

	printf ("\nEnter the elements of Array A ");
	for (int i=0; i<sizeA; i++)
	scanf(" %d", &a[i]);

	printf("\nPlease enter the size of Array B " );
	scanf(" %d", &sizeB);
	int b[sizeB];

	printf ("\nEnter the elements of Array B ");
	for (int i=0; i<sizeB; i++)
	scanf(" %d", &b[i]);

	asc(&a[0],sizeA);

	asc(&b[0],sizeB);

	int sizeC= (sizeof(a)+sizeof(b))/4;
	int c[sizeC];

	int i=0;
	for(i=0; i<sizeA; i++)
	{
		c[i]=a[i];
	}
```

```
	for (i=0; i<sizeB; i++)
	c[i+sizeA]=b[i];


	asc(&c[0], sizeC);

		for (int i=0; i<sizeC;i++)
	printf(" %d", c[i]);
}

int asc(int a[], int size) //function for ascending order
{


	int temp;

	for (int i=0; i<size; i++)
	{
		for (int j=0; j<size-1; j++)
		{
			if (a[j]>a[j+1])
			{
				temp=a[j];
				a[j]=a[j+1];
				a[j+1]=temp;
			}
		}
	}
		return &a[0];
}
```

Output:



Task 04:

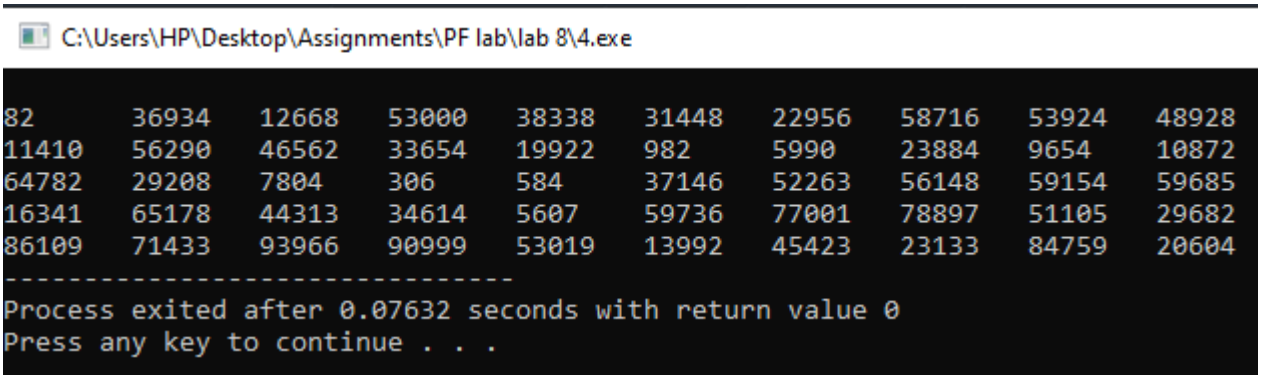
Write a C program that declares an array Numbers of 50 components of type int. Initialize the array so that the first 25 components are divisible by 2, and the last 25 components are divisible by 3.

Output the array so that 10 elements per line are printed.

Program:

```
#include <stdio.h>
int main()
{

int rn[50]; //randomn numbers

for (int i=0; i<50; i++)
{          if (i<25)
rn[i]=rand()*2;
else
rn[i]=rand()*3;
}

for (int i= 0; i<50; i++)
{
if(i%10==0)
printf("\n");

printf("%d\t", rn[i]);
}
}
```

Output:



Task 05:

The teacher at a university needs help in grading a True/False test. The test contains the students' IDs and test answers in the form:
1022 TFTTF TFTT where the student ID is 1022 and the answer to question 1 is True, the answer to question 2 is False, and so on. This student did not answer question 6 so a blank in written. The exam has 0 questions, and the class has 15 students. Each correct answer is awarded one points and no point are awarded for wrong answers and answers that are left blank. Write a program that processes the data. The output should be the student's ID, followed by the answers, followed by the test score, followed by the test grade. Assume the following grade scale:

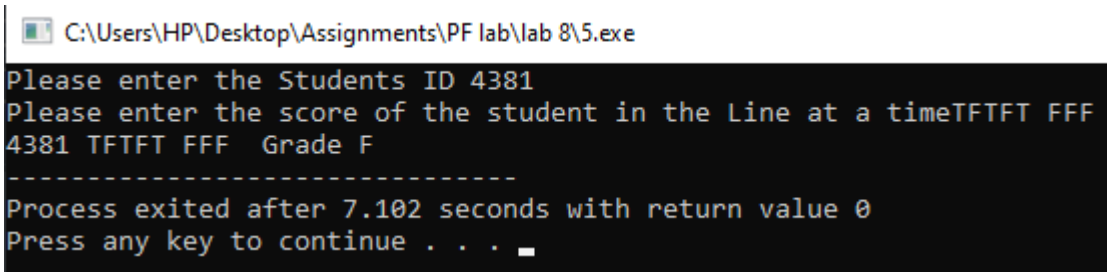
- 90%–100% - A
- 80%–89.99% - B 70%–79.99% - C
- 60%–69.99% - D
- 0%–59.99% - F

Program:

```
#include <stdio.h>
int main()
{
	int id;
	printf("Please enter the Students ID ");
	fflush(stdin);
	scanf("%d", &id);

	char score[10];
	printf("Please enter the score of the student in the Line at a time");
	fflush (stdin);
	gets(score);
	printf("%d ", id);
	int t=0;
	for (int i=0; i<10; i++)
	{
		printf("%c", score[i]);
		if (score[i]=='T')
		t++;
	}

	float p= t*100/10; //percent

	if (p<=100&&p>=90)
	printf(" Grade A");

	if (p<90&&p>=80)
	printf(" Grade B");
```

```
        if (p<80&&p>=70)
        printf(" Grade C");

        if (p<70&&p>=60)
        printf(" Grade D");

        if (p<60&&p>=0)
        printf(" Grade F");
}
```

Output:



Task 06:

Write a program that takes a non-negative integers representing an elevation map where the width of each bar is 1, compute how much water it can trap after raining.

Example 1:


```
Input: height = [0,1,0,2,1,0,1,3,2,1,2,1]
Output: 6
Explanation: The above elevation map (black section) is represented by array
[0,1,0,2,1,0,1,3,2,1,2,1]. In this case, 6 units of rain water (blue section)
are being trapped.
```

Example 2:

```
Input: height = [4,2,0,3,2,5]
Output: 9
```

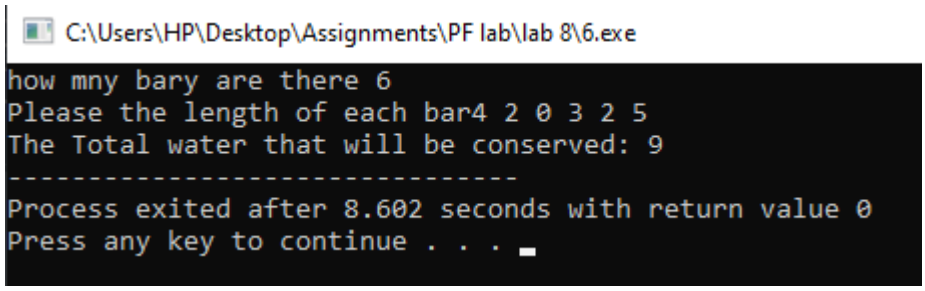
Program:

```
#include <stdio.h>
#include <string.h>
int main(){

int s;
printf("how mny bary are there ");
scanf("%d", &s); // size

int bar[s];

printf("Please the length of each bar");
for (int i=0; i<s; i++)
scanf("%d", &bar[i]);

int l[s];
l[0]=bar[0];
int max= bar[0];
```

```
for (int i=1; i<s; i++)
{
if (bar[i]>max)
{
        max=bar[i];
}
l[i]= max;
}

int r[s];
r[s-1]= bar[s-1];
max= bar[s-1];
for (int i=s-2; i>=0; i--)
{
if (bar[i]>max)
{
        max= bar[i];
}
r[i]=max;
}

int d[s]; //difference

for (int i=0; i<s; i++)
{
if (l[i]<r[i] || l[i]==r[i])
d[i]= l[i]-bar[i];

else
d[i]= r[i]-bar[i];
}
int sum=0;
for (int i=0; i<s; i++)
```

```
sum=sum+d[i];


printf("The Total water that will be conserved: %d", sum);


}
```

Output:



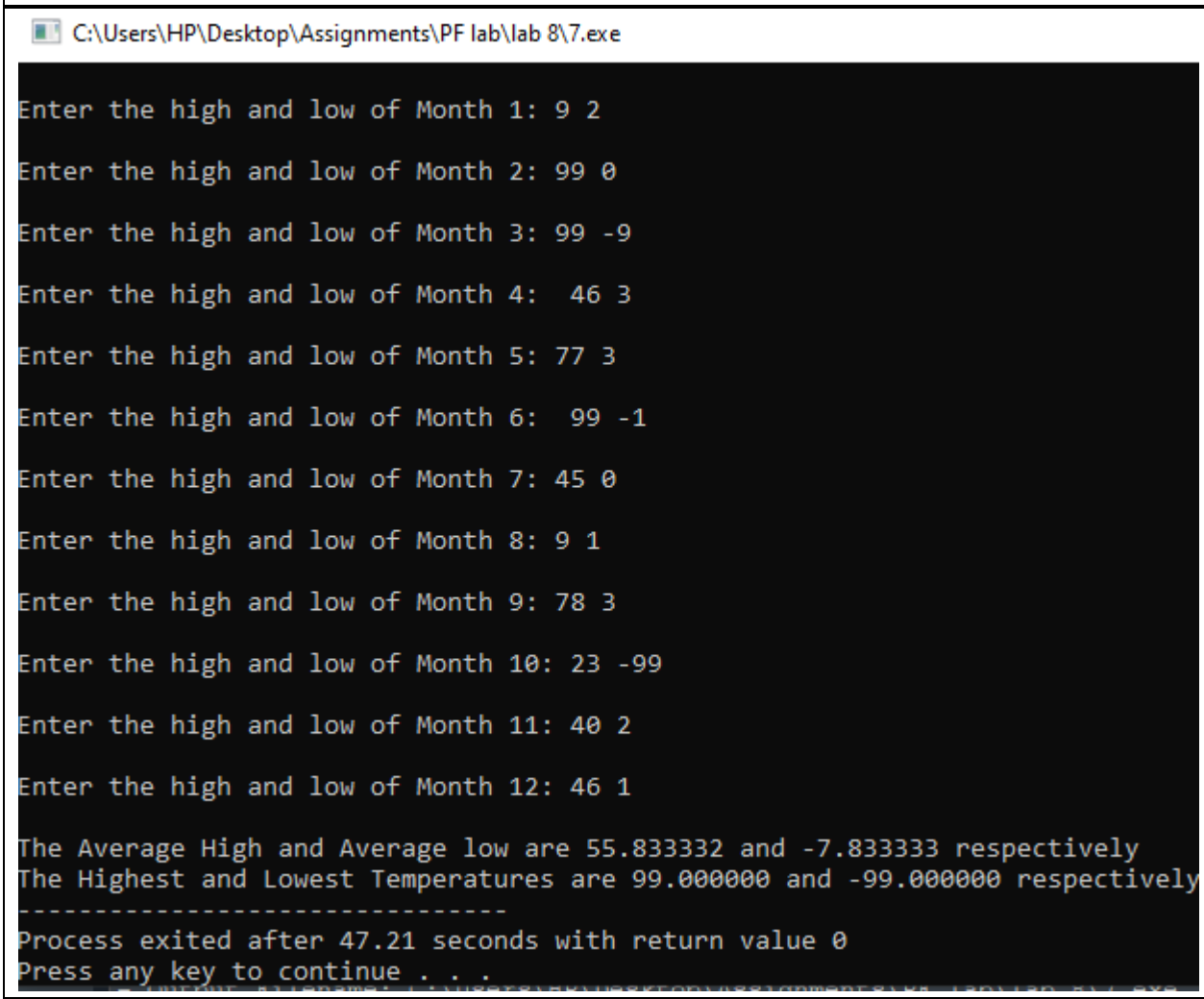
Task 07:

Write a program that uses a two-dimensional array to store the highest and lowest temperatures for each month of the year. The program should output the average high, average low, and the highest and lowest temperatures for the year.

Program:

```
#include <stdio.h>
int main()
{
float a[12][2]; //temperature
float hsum=0, lsum=0;
float max, min;

for (int i=0; i<12; i++)
{
printf("\nEnter the high and low of Month %d: ", i+1);

for (int j=0; j<2; j++)
{
        scanf("%f",&a[i][j]);

        if (i==0 && j==0)
        max=a[i][j];

        if(i==0 &&j==1)
        min=a[i][j];

        if (j==0)
                {
                        hsum=hsum+a[i][j];
                        if (a[i][j]>max)
                                max= a[i][j];
                }
        if (j==1)
                {
                        lsum=lsum+a[i][j];
                        if(a[i][j]<min)
                                min=a[i][j];
                }
}

}
printf("\nThe Average High and Average low are %f and %f respectively", hsum/12, lsum/12);
printf("\nThe Highest and Lowest Temperatures are %f and %f respectively", max, min);
```

}
Output:



Task 08:

Create a program that take input from user (Number, Asterisk (*) and Alphabet) and design the following patterns based on user input.

```
1          *          A
12         **         AB
123        ***        ABC
1234       ****       ABCD
12345      *****      ABCDE
```

You should also take input for number of rows from user.

Sample output:

```
Enter the pattern you want to create
N for number, S for star and A for alphabet: A

Enter number of rows you want your pattern to be: 3

A
AB
ABC
```

Program:

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>

int main()
{
	char choice;
	printf("Enter the pattern you want to create \nN for
Number, S for Star and A for Alphabet: ");
	scanf("%c", &choice);

	int row;
	printf("Enter the number of rows you want your
pattern to be: ");
	scanf("%d", &row);

	switch (choice)
	{
		case 'N':

			for (int i= 1; i<=row; i++)
			{
				for (int j=1; j<=i; j++)
```

```
                                {
                                        printf("%d", j);
                                }
                                printf("\n");
                        }
                        break;

                case 'S':
                        for (int i=1; i<=row; i++)
                        {
                                for (int j=1; j<=i; j++)
                                {
                                        printf("*");
                                }
                                printf("\n");
                        }
                        break;
                case 'A':
                        for (int i='A'; i<'A'+row; i++)
                        {
                                for (int j= 'A'; j<=i; j++)
                                {
                                        printf("%c", j);
                                }
                        printf("\n");
                        }
                        break;
        }
}
```

Output:

```
C:\Users\HP\Desktop\Assignments\PF lab\lab 8\8.exe
Enter the pattern you want to create
N for Number, S for Star and A for Alphabet: A
Enter the number of rows you want your pattern to be: 9
A
AB
ABC
ABCD
ABCDE
ABCDEF
ABCDEFG
ABCDEFGH
ABCDEFGHI

--------------------------------
Process exited after 8.617 seconds with return value 0
Press any key to continue . . .
```

Task 09:

Create a program which adds or subtracts matrices upon user desire. It can add/subtract 2, 3 and 4 3x3 matrices together. Write a program that takes user input:

- the operation the user wants to perform (add/subtract)
- how many matrices the user wants to add/subtract (2, 3 or 4) ask user to enter

elements of all the matrices print the result in the following format:

```
What operation you want to perform
1. Addition
2. Subtraction

1

Select number of matrices:
1. 2 matrices
2. 3 matrices
3. 4 matrices

2

Enter elements of matrix A: 2 2 2 2
Enter elements of matrix B: 1 1 1 1

2  2  2  +  1  1  1  =  3  3  3
1  1  1     1  1  1     2  2  2
1  2  3     1  2  3     2  4  6
```

Also check for valid input. A user cannot enter other choices except for the given choices. If he selects any other option, it should print an error message.

Program:

```
#include <stdio.h>
int main()
{
	int op;//operand
	printf("What operation you want to perform?");
	printf("\n1. Addition \n2. Subtraction\n\n");
	scanf("%d", &op);

	if (op==1 || op==2)
	{
		int dim; //dimension
		printf("\nSelect the number of Matrices");
		printf("\n1. 2 Matrices \n2. 3 Matrices \n3. 4 Matrices\n");
		scanf("%d", &dim);
```

```
if (dim==1 || dim==2 ||dim==3)
{
	dim++;
	int m1[dim][dim], m2[dim][dim]; //both matrices
	printf("\nEnter Elements of Matrix A: ");
	for (int i=0; i<dim; i++)
		{
		for (int j=0; j<dim; j++)
		scanf(" %d", &m1[i][j]);
		}

	printf("\nEnter Elements of Matrix B ");
	for (int i=0; i<dim; i++)
		{
		for (int j=0; j<dim; j++)
		scanf(" %d", &m2[i][j]);
		}
	int r[dim][dim]; //resultant matrix

	for (int i=0; i<dim; i++)
	{
		for (int j=0; j<dim; j++)
		{
			if (op==1)
				r[i][j]= m1[i][j] + m2[i][j];
			if (op==2)
				r[i][j]= m1[i][j]-m2[i][j];
		}
	}

	for (int i=0; i<dim; i++)
	{
	for (int j=0; j<dim; j++)
		{
			printf("%d ", m1[i][j]);
		}
		if (i==0)
		{
			if (op==1) printf("+ ");
			if (op==2) printf("- ");
		}
		else printf("  ");

	for (int k= 0; k<dim; k++)
		{
			printf("%d ", m2[i][k]);
```

```
                                }
                                if (i==0) printf("= ");
                                else printf("  ");

                        for (int l=0; l<dim; l++)
                                {
                                        printf("%d ", r[i][l]);
                                }
                        printf("\n");
                        }
                }
                else printf("\nError");
        }
        else printf("\nError");
}
```

Output: