

Midterm 2

4th Nov 2019, 11:00 am – 12:00 noon

Course Code: CS118	Course Name: Programming Fundamentals
Instructor Name: Dr. Farooque / Sir Shahzad / Sir Shoaib / Sir Basit	
Student Roll No: Solution By Basit Jasani	Section No:

Instructions:

- Return the question paper and make sure to keep it inside your answer sheet.
- Read each question completely before answering it. There are **5 questions and 2 pages**.
- In case of any ambiguity, you may make assumption. But your assumption should not contradict any statement in the question paper.
- You are **not allowed to write** anything on the question paper (except your ID and group).

Time: 60 minutes.

Max Marks: 50 Marks

Q 1. The results from the mayor's race have been reported by each area as follows:

[Marks 5+5+5]

Area	Candidate 1	Candidate 2	Candidate 3	Candidate 4
1	192	48	206	37
2	147	90	312	21
3	186	12	121	38
4	114	21	408	39
5	267	13	382	29

Your task is to store the above data into a 2D matrix to perform the analysis given below.

```

int main()
{
    int a[5][4] = {
        {192, 48, 206, 37},
        {147, 90, 312, 21},
        {186, 12, 121, 38},
        {114, 21, 408, 39},
        {267, 13, 382, 29}
    };

    int m;
    Check_Winner1(a);
    Check_Winner2(a);
    m = Check_Winner3(a);
    printf("\n%d", m);
}

```

- a) Write a function named “**Check_Winner1**” that would receive the 2D matrix as its input and display the winner by each area.

Expected output:

Area 1	:	Candidate 3
Area 2	:	Candidate 3
Area 3	:	Candidate 1
Area 4	:	Candidate 3
Area 5	:	Candidate 3

```
Check_Winner1(int a[5][4])
{
    int i,j, output,max;

    for (i=0; i<=4; i++)
    {
        max = 0, output = 0;
        for(j=0; j<=3; j++)
        {
            if(max<a[i][j])
            {
                max = a[i][j];
                output = j+1;
            }
        }
        printf("\nArea %d: Candidate %d\n", i+1, output);
    }
}
```

- b) Write a function named “**Check_Winner2**” that would receive the 2D matrix as its input and display the total votes received by each candidate.

Expected output:

```
Candidate 1 : Total votes 906
Candidate 2 : Total votes 184
Candidate 3 : Total votes 1429
Candidate 4 : Total votes 164
```

```
Check_Winner2(int a[5][4])
{
    int i,j,sum;
    for (j=0; j<=3; j++)
    {
        sum = 0;
        for(i=0; i<=4; i++)
        {
            sum = sum + a[i][j];
        }
        printf("\n\nCandidate %d: %d\n", j+1, sum);
    }
}
```

- c) Write a function named “**Check_Winner3**” that would receive the 2D matrix as its input and returns the area number in which maximum votes are casted.

Expected return: 5

```
int Check_Winner3(int a[5][4])
{
    int sum, largest=0, k = 0, i, j;
    for (i=0; i<=4; i++)
    { sum = 0;
        for(j=0; j<=3; j++)
        {
            sum = sum + a[i][j];
        }
        if(sum>largest)
        {
            largest = sum, k =
i+1;
        }
    }
    return k; }
```

- Q 2.** Write a function named "**rotate_left(int arr[], int arrSize)**" that would do the task mentioned below.

[Marks 5]

The function should shift the contents of each cell one place to the left, except for the contents of the first cell, which should be moved into the last. Thus, for example, if the array passed to the function looks like this:

[5, 3, 7, 8, 1, 4]

Then function would update the array like this:

[3, 7, 8, 1, 4, 5]

```
rotate_left(int a[], int n)
{
    int i, temp = a[0];
    for (i=0; i<n-1; i++)
    {
        a[i] = a[i+1];
    }
    a[n-1] = temp;
    for (i=0; i<n; i++)
    {
        printf("%d\n", a[i]);
    }
}

int main()
{
    int a[6] = {5,3,7,8,1,4};
    rotate_left (a, 6);
    getchar();
    return 1;
}
```

- Q 3.** Write a program to create a character linear array of N elements. Write following functions and pass this array to the functions: *[Marks 5+5]*

```
int main()
{
    char arr[]="Hi, My name is Basit. I am teaching PF.";
    int size;
    size = strlen(arr);
    int count = WordCap (arr, size);
    printf("\nThe count of words is %d\n", count);
    DelPunctuation(arr, size);
    getchar();
    return 1;
}
```

- a) **int WordCap(char arr[], int size):** Count and print in “main()” the number of words in an array and capitalize each word’s first letter.

```
int WordCap (char arr[], int size)
{
    int count = 0, i;
    for(i=0; i<size; i++)
    {
        if(i==0)
        {
            if((arr[i]>='a' && arr[i]<='z'))
            { arr[i]=arr[i]-32; count++; }
            continue;
        }
        if(arr[i] == ' ')
        {
            ++i; count++;
            if(arr[i]>='a' && arr[i]<='z')
            {
                arr[i]=arr[i]-32; continue;
            }
        }
        else
    }
```

```

        {
            if(arr[i]>='A' && arr[i]<='Z')
            {
                arr[i]=arr[i]+32; count++;
            }
        }
    }
    for (i=0; i<size; i++)
    {
        printf("%c", arr[i]);
    }
    return count;
}

```

- b) **Void DelPunctuation(char arr[], int size):** Delete spaces, comma, apostrophe and full stop.

```

DelPunctuation(char arr[], int size)
{
    int i, j=0; char arr2[100];
    for(i=0; i<size; i++)
    {
        if (arr[i] == ' ' || arr[i] == ',' || arr[i] == '?' || arr[i] == '.')
            continue;
        else
        {
            arr2[j] = arr[i];
            j++;
        }
    }

    strcpy(arr, arr2);
    int len = strlen(arr);
    for (i=0; i<len; i++)
    {
        printf("%c", arr[i]);
    }
}

```

Q 4. Write a structure to store the roll no., score, name, age and address of 100 students.

Score is another user defined structure which has member's subject1, subject2 and subject3. Store the information of the students. *[Marks 5+5]*

- Write a function to print the names of all the students having age between 18 and 22.
- Write another function to display the details of the student whose roll no is given (i.e. roll no. entered by the user).

Method 1

```
struct score
{
    int sub1; int sub2; int sub3;
};

struct student
{
    int rollnum; char name[100]; int age; char
address[100]; struct score s;
};

Function1 (struct student st[100])
{
    int i;
    for(i=0; i<100; i++)
    {
        if (st[i].age > 18 && st[i].age < 22)
        {
            puts(st[i].name);
        }
    }
}

Function2(struct student st[100])
{
    int i,x;
    printf("Enter the student number you want to
search?");
    scanf("%d", &x);

    for(i=0; i<100; i++)
    {
        if (st[i].rollnum == x)
        {
```

```

                printf("%s", st[i].name); printf("\n
%d", st[i].s.sub1); printf("\n %d", st[i].s.sub2); printf("\n
%d", st[i].s.sub3);
            }
        }
    }

int main()
{
    struct student st[100];
    int i;
    for(i=0; i<100; i++)
    {
        //hardCoded
        st[i].rollnum = i*2;
        strcpy (st[i].name , "K");
        st[i].age = i;
        strcpy (st[i].address, "HardCoded");
        st[i].s.sub1 = i+4;
        st[i].s.sub2 = i+8;
        st[i].s.sub3 = i+12;
    }
    Function1(st);
    Function2(st);

    getchar();

}

```


Method 2

```
struct score
{
    int sub1; int sub2; int sub3;
};
```

```
struct student
{
    int rollnum; char name[100]; int age; char
address[100];
};
```

```
Function1 (struct student s[100])
{
    int i;
    for(i=0; i<100; i++)
    {
        if (s[i].age > 18 && s[i].age < 22)
        {
            puts(s[i].name);
        }
    }
}
```

```
Function2(struct student s[100], struct score sc[100])
{
    int i,x;
    printf("Enter the student number you want to
search?");
    scanf("%d", &x);

    for(i=0; i<100; i++)
    {
        if (s[i].rollnum == x)
        {
            printf("%s", s[i].name); printf("\n %d",
sc[i].sub1); printf("\n %d", sc[i].sub2); printf("\n %d",
sc[i].sub3);
        }
    }
}
```

```

    }
}

int main()
{
    struct student s[100];
    struct score sc[100];
    int i;
    for(i=0; i<100; i++)
    {
        //hardCoded
        s[i].rollnum = i*2;
        strcpy (s[i].name , "K");
        s[i].age = i;
        strcpy (s[i].address, "HardCoded");
        sc[i].sub1 = i+4;
        sc[i].sub2 = i+8;
        sc[i].sub3 = i+12;
    }

    Function1(s);
    Function2(s, sc);

    getchar();
    return 1;
}

```

Q 5. A word is considered as 'teen' if the word contains letters 't', and 'e' in any order. For example, we would say that the following words are “teen”: tiger, tired, stress and teenager, because they each contain letters 't' and 'e'. Write a recursive function “teen(char a[])”, that will print 'teen', if the given word contains letter 't' and 'e' else print 'not teen'. *[Marks 10]*

Expected Input	:	tiger
Expected Output	:	teen
Expected Input	:	smart
Expected Output	:	not teen

Method 1

```
fun1(char a[],int i)
{
    static int t,e;

    if(a[i]=='\0')
    {
        if((t>=1)&&(e>=1))
        {
            printf("teen");
        }
        else
        {
            printf("not teen");
        }
    }
    else
    {
        if(a[i]=='t')
        {
            t++; i++;
            fun1(a,i);
        }
        else if(a[i]=='e')
        {
            e++; i++;
            fun1(a,i);
        }
        else
        {
            i++;
            fun1(a,i);
        }
    }
}
```

```

    }
}

int main()
{
    int l,i=0;
    char a[100];
    printf("enter a string\n");
    scanf("%s",a);

    fun1(a,i);

    getchar();

}

```

Method 2

```

fun1(char a[],int i, int t, int e)
{
    if(a[i]=='\0')
    {
        if((t>=1)&&(e>=1))
        {
            printf("teen");
        }
        else
        {
            printf("not teen");
        }
    }
    else
    {
        if(a[i]=='t')
        {
            t++;
            i++;
            fun1(a,i,t,e);
        }
        else if(a[i]=='e')

```

```

        {
            e++;
            i++;
            fun1(a,i,t,e);
        }
        else
        {
            i++;
            fun1(a,i,t,e);
        }
    }

int main()
{
    int l,i=0;
    char a[100];
    printf("enter a string\n");
    scanf("%s",a);

    fun1(a, 0, 0, 0);
    //printf("%d",l);
    getchar();

}

```

BEST OF LUCK!