File Input/Output Exercises

1. **A program to show how you can send data to a file.**

   Data can be send to a file using function fprintf(). **See, how similar is function** fprintf() **to function** printf().

```
#include<stdio.h>

void main()
{
      FILE *fp;
      fp = fopen("output1.txt","w+");

      printf("Programming \n");     // printf()is sending data to a screen/console.
      fprintf(fp,"Something different, let's see \n");     // fprintf() is sending data to a file.
}
```

2. **A program to show how you can get inputs/data from a file.**

   Data can be obtained from a file using a function fscanf()**. Again see, how similar is function** fscanf() **to function** scanf().

```
#include <stdio.h>

void main()
{
      int phy, che, com, p, c, cs;

      FILE *fp;
      fp = fopen("input2.txt","r+");

      printf("Enter mark of 3 subjects: ");
      scanf("%d%d%d",&phy,&che,&com);     // scanf() is getting inputs/data from a screen/console.

      fscanf(fp,"%d%d%d",&p,&c,&cs);     // fscanf() is getting inputs/data from a file.

      printf("Data from file are: \n");
      printf("\n%d  %d  %d \n",p,c,cs);
}
```

3. An example to show different file opening modes.

There are 6 possible file opening modes.

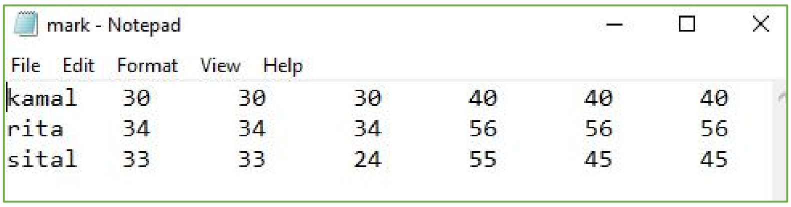
Opening Mode	Reading data from file	Writing data to file	If file does not exist	While opening, data in file
r	Allowed	Not allowed	File not created	Not deleted
w	Not allowed	Allowed	File is created	Deleted
r+	Allowed	Allowed	File not created	Not deleted
w+	Allowed	Allowed	File is created	Deleted
a	Not allowed	Allowed	File is created	Not deleted, data added to end
a+	Allowed	Allowed	File is created	Not deleted, data added to end

```
#include <stdio.h>

void main()
{
    int phy, che, com, p, c, cs;

    FILE *fp;
    fp = fopen("input5.txt","r");

    //It's a good idea to check whether a file has successfully opened.
    if(fp==NULL)
    {
            printf("Couldn't open file. \n");
            exit(1);
    }

    printf("Enter mark of 3 subjects: ");
    scanf("%d%d%d",&phy,&che,&com);
    fprintf(fp,"%d  %d  %d\n",phy,che,com);

    rewind(fp);
    fscanf(fp,"%d%d%d",&p,&c,&cs);
    printf("\n%d  %d  %d",p,c,cs);
}
```

Change opening mode to all 6 types, see what output do you get?

4. **A program to show how to find whether you have reached the end of a file while reading data and therefore stop reading data from a file.**

>> For this exercise, you should create a file called *mark.txt* with some data, like the one shown below, before you run the program.



```
kamal   30      30      30      40      40      40
rita    34      34      34      56      56      56
sital   33      33      24      55      45      45
```

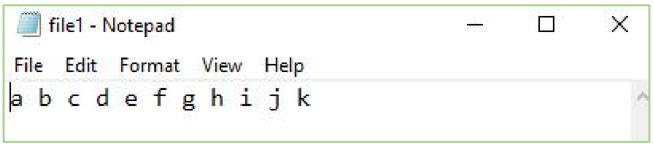
```
#include <stdio.h>

int main()
{
    char name[10];
    int mark[6], sum, percent, i, j;

    FILE *fp;
    fp = fopen("mark.txt","r+");

    while(feof(fp)==0)   //Function feof() helps to detect whether you've reached the end of a file.
    {
            fscanf(fp,"%s",name);

            i=0;
            while(i<6)
            {
                    fscanf(fp,"%d",&mark[i]);
                    i++;
            }

            sum=0;
            i=0;
            while(i<6)
            {
                    sum = sum + mark[i];
                    i++;
            }
            percent = sum/6;

            printf("%s \t",name);
            printf("%d \n",percent);
    }
}
```

5. A program to demonstrate how you can randomly read/write data from/to a file.

- A computer maintains the current file reading/writing position. Data is always read/write from/to this position. Note once the data is read/write from/to the current position, the file reading/writing position is moved one position ahead.
- In the beginning, the file reading/writing position is at the beginning of a file, i.e., at 0. However, a position can be changed using the function fseek(). It is of the form *fseek(fp, offset, position)* where *offset* is the number of position (bytes) to be moved from *position* which takes values: 0 (to indicate beginning of file), 1 (current position) or 2 (end of a file).
- The function ftell() tells the current file reading/writing position.
- The file reading/writing position can be set to 0 any time if required using the function rewind().

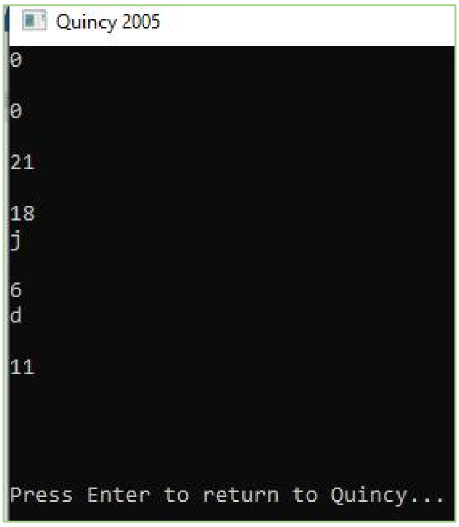
```
#include <stdio.h>

void main()
{
    int position;   char data;

    FILE *fp;
    fp = fopen("file1.txt","r+");

    rewind(fp);
    position = ftell(fp);
    printf("%d \n\n",position);

    fseek(fp,0,0);
    position = ftell(fp);
    printf("%d \n\n",position);

    fseek(fp,0,2);
    position = ftell(fp);
    printf("%d \n\n",position);

    fseek(fp,-3,2);
    position = ftell(fp);
    printf("%d \n",position);
    fscanf(fp,"%c",&data);
    printf("%c \n\n",data);

    fseek(fp,6,0);
    position = ftell(fp);
    printf("%d \n",position);
    fscanf(fp,"%c",&data);
    printf("%c \n\n",data);

    fseek(fp,4,1);
    position = ftell(fp);
    printf("%d \n\n",position);
    fscanf(fp,"%c",&data);
    printf("%c \n\n",data);
}
```

>> Let's say *file1.txt* contains the following data,



```
a b c d e f g h i j k
```

>> When you run this program, you will get:



```
0

0

21

18
j

6
d

11




Press Enter to return to Quincy...
```

>> Can you explain why do you get above outputs?

6. A program to show the use of function putw(), putc(), getw() **and** get()**.**

```
#include <stdio.h>

int main()
{
	int id, i;
	char grade;

	FILE *fp;
	fp = fopen("result.txt","w+");

	for(i=0;i<3;i++)
	{
			printf("Enter id: ");
			scanf("%d",&id);
			putw(id,fp);	//Allows to put an integer in a file. Note, this function often does not properly.
			fscanf(fp,"\t");

			fflush(stdin);
			printf("Enter grade: ");
			scanf("%c",&grade);
			putc(grade,fp);	//Allows to put a character in a file.
			fprintf(fp,"\n");
	}

	printf("\n");
	rewind(fp);
	for(i=0;i<3;i++)
	{
			id = getw(fp);	//Allows us to get an integer from a file. Note, this function often does not properly.
			printf("%d \t",id);
			grade = getc(fp);	//Allows us to get a character from a file.
			printf("%c \n",grade);
	}
}
```

7. A complete program to show the file input/output feature.

```
#include <stdio.h>

int main()
{
    char name[10];
    int mark[6], sum, percent, i, j;

    FILE *fp;
    fp = fopen("mark.txt","w+");

    if(fp==NULL)
    {
            printf("Couldn't open file. \n");
            exit(1);
    }

    j=0;
    while(j<3)
    {
            //Getting data from a screen/console.
            printf("Enter name: ");
            scanf("%s",name);

            i=0;
            while(i<6)
            {
                    printf("Enter mark: ");
                    scanf("%d",&mark[i]);
                    i++;
            }
            printf("\n");

            //Sending data to a file.
            fprintf(fp,"%s",name);
            fprintf(fp,"\t");

            i=0;
            while(i<6)
            {
                    fprintf(fp,"%d",mark[i]);
                    fprintf(fp,"\t");    //Can you tell why we are sending \t to a file?
                    i++;
            }
            fprintf(fp,"\n");
            j++;
    }
```

```
        printf("\n");
        rewind(fp);
        j=0;
        while(j<3)
        {
                //Getting data from a file.
                fscanf(fp,"%s",name);

                i=0;
                while(i<6)
                {
                        fscanf(fp,"%d",&mark[i]);
                        i++;
                }

                //Doing processing (finding percent here) and displaying the result.
                sum=0;
                i=0;
                while(i<6)
                {
                        sum = sum + mark[i];
                        i++;
                }
                percent = sum/6;

                printf("%s \t",name);
                printf("%d \n",percent);
                j++;
        }

    }
```