Chapter 11

Files

Answers to Review Questions

1. Refer Section 11.1 of Chapter 11.
2. Refer Section 11.1 of Chapter 11.
3. Refer Section 11.1 of Chapter 11.
4. Refer Section 11.1.3 of Chapter 11.
5. Refer Table 11.1 of Chapter 11.
6. Refer Section 11.2.2 of Chapter 11.
7. Refer Section 11.2.3 of Chapter 11.
8. When the *fclose*() is executed any unwritten buffered data for the stream will be written to the file and any unread buffered data will be discarded.
9. Refer Section 11.3.2 of Chapter 11.
10. When an output stream is un-buffered, information appears on the destination device as soon as it is written. When it is buffered, characters are saved internally and then written out as a group.
11. Refer Section 11.8.2 of Chapter 11.
12. When we have to deal with text files for which position of the data cannot be calculated.
13. Refer Sections 11.8.4 and 11.8.5 of Chapter 11.
14. Refer Section 11.9 of Chapter 11.
15. Refer Section 11.1.2 of Chapter 11.
16. Refer Sections 11.3 and 11.4 of Chapter 11.
17. a. Refer Section 11.2.2. b. Refer Section 11.2.3. c. Refer Section 11.6. d. Refer Section 11.6.1.
18. Refer Section 11.5 of Chapter 11.
19. Refer Section 11.5 of Chapter 11.
20. The file pointer is positioned at the end of the file. Thus you need to reset the file pointer so that it points to the desired location in the file.
21. Refer Section 11.6 of Chapter 11.
22. Refer Section 11.8.3 of Chapter 11.
23. Refer Section 11.8.3 of Chapter 11.
24. If a file's buffer has to be flushed without closing it then use fflush() or flushall() to flush the buffers of all open streams.

26.
```
#include <stdio.h>
#include <stdlib.h>
#define FNAME "INTEGERS.TXT"
#define FILE1 "EVEN.TXT"
#define FILE2 "ODD.TXT"
int main()
{   FILE *fp, *fp1, *fp2;
        int num;
  fp = fopen(FNAME, "wb");
  if (fp == NULL)
  {     printf("Error opening file %s\n", FNAME);
            exit(1);
  }
  fp1 = fopen(FILE1, "wb");
  if (fp1 == NULL)
```

Oxford University Press

Copyright© 2018 by Oxford University Press

```
        {     printf("Error opening file %s\n", FILE1);
                exit(1);
        }
        fp2 = fopen(FILE2, "wb");
        if (fp2 == NULL)
        {     printf("Error opening file %s\n", FILE2);
              exit(1);
        }
  while(1)
        {       printf("\n Enter the number : ");
                scanf("%d", &num);
                putw(num,fp);
                if(num==-1)break;
     }
        fclose(fp);
        fp = fopen(FNAME, "rb");
        if (fp == NULL)
        {     printf("Error opening file %s\n", FNAME);
                exit(1);
        }
        while(1)
        {     num = getw(fp);
          if(num==-1)   break;
          else if(num%2==0)
                        putw(num,fp1);
          else putw(num,fp2);
        }
        fcloseall();
      }
```

Answers to Programming Exercises

1.
```
#include<stdio.h>
#include<stdlib.h>

int main()
{
FILE *fp = tmpfile();
char feedback[100];
if(fp==NULL)
{       printf("\n The file could not be opened");
        exit(1);
}
printf("\n Kindly give the feedback on this book : ");
gets(feedback);
fflush(stdin);
fputs(feedback, fp);
printf("\n Data Stored in the file");
  fclose(fp);
}
```

2.

Oxford University Press

Copyright© 2018 by Oxford University Press

```
#include<stdio.h>
#include<stdlib.h>
int main()
{       typedef struct employee
        {       int emp_code;
                char name[20];
                int hra;
                int da;
                int ta;
        };
        FILE *fp;
        struct employee e, emp[5];
        int i, found=0,empno;
        float sal;
        fp = fopen("employee.txt", "w");
        if(fp == NULL)
        {       printf("\n File Opening Error");
                return 0;
        }
        for(i=0;i<5;i++)
        {       printf("\n Enter the employee number : ");
                scanf("%d", &emp[i].emp_code);
                printf("\n Enter the name : ");
                scanf("%s", emp[i].name);
                printf("\n Enter the hra, da and ta : ");
                scanf("%d %d %d", &emp[i].hra, &emp[i].da, &emp[i].ta);
        }

       // PRINT ON SCREEN
        for(i=0;i<5;i++)
        {       printf("\n ********EMPLOYEE'S DETAILS *******");
                printf("\n EMPLOYEE No. = %d", emp[i].hra);
                printf("\n NAME. = %s", emp[i].name);
                printf("\n HRA DA TA = %d %d %d", emp[i].hra, emp[i].da, emp[i].ta);

// WRITE TO FILE
                fprintf(fp,"%d %s %d %d %d ", emp[i].emp_code, emp[i].name, emp[i].hra,
emp[i].da, emp[i].ta);
        }

        fclose(fp);

        fp =  fopen("employee.txt", "r");
        if(fp==NULL)
        {       printf("\n Error opening file");
                exit(1);
        }

        printf("\n Enter the employee no. of the employee whose salary has to be calculated : ");
        scanf("%d", &empno);
        for(i=0;i<5;i++)
        {       fscanf(fp, "%d %s %d %d %d", &e.emp_code,  e.name, &e.hra, &e.da, &e.ta);
                if(e.emp_code == empno)
                {       found = 1;
                printf("\n The details of student are - ");
                printf("%d %s %d %d %d", e.emp_code,  e.name, e.hra, e.da, e.ta);
```

Oxford University Press

Copyright© 2018 by Oxford University Press

```
                    sal = e.hra + e.da + e.ta;
                    printf("\n SALARY = %d", sal);
                }
            }
                if (found==0)
                    printf("\n Record not found in the file");
                fclose(fp);
            }
```

3.

```
    while(1)
            {
                printf("\n Enter the employee number : ");
                scanf("%d", &empno);
                if(empno == -1)
                        break;
                fscanf(fp, "%d %s %d %d %d", &e.emp_code,  e.name, &e.hra, &e.da, &e.ta);
                if(e.emp_code == empno)
                {       found = 1;
                printf("\n The details of employee are - ");
                printf("%d %s %d %d %d", e.emp_code,  e.name, e.hra, e.da, e.ta);
                }
            }
```

14.
```
#include<stdio.h>
#include<string.h>
int main(int argc, char *argv[])
{       FILE *fp1, *fp2;
        char str[20];
        if(argc != 3)
        {       printf("\n Full information is not provided");
                return 0;
        }
        fp1 = fopen(argv[1], "w");
        if(fp1 == NULL)
        {       printf("\n File Opening Error");
                return 0;
        }
        strcpy(str, argv[2]);
        fprintf(fp1, "%s", str);
        fclose(fp2);
}
```

19. Open the file in a while loop and use a variable count that is incremented each time the user enters a filename. If count becomes 3 exit from the program else do the processing

20.
```
#include<stdio.h>
        #include<string.h>
        #include<stdlib.h>
        #define FILE1 "ACT.TXT"
        #define FILE2 "DANCE.TXT"
        #define FILE3 "NAMES.TXT"
```

Oxford University Press

Copyright© 2018 by Oxford University Press

```
int main()
{       FILE *fp1, *fp2, *fp3;
        char str[20], names_act[5][10], names_dance[5][10],names[10][10],temp[20];
        int i=0,n,j=0;
        fp1 = fopen(FILE1, "wb");
        if(fp1==NULL)
        {       printf("\n The file could not be opened");
                exit(1);
        }
        printf("\n Enter the name of students in ACtivity1 : ");
        while(1)
        {       gets(names_act[i]);
                if(strcmp(names_act[i],".")==0)break;
                fwrite(names_act[i], strlen(names_act[i]),1,fp1);
                i++;
                fflush(stdin);
        }
        fp2 = fopen(FILE2, "wb");
        if(fp2==NULL)
        {       printf("\n The file could not be opened");
                exit(1);
        }
        i=0;
        printf("\n Enter the name of students in ACtivity 2 : ");
        while(1)
        {       gets(names_dance[i]);
                if(strcmp(names_dance[i],".")==0)break;
                fwrite(names_dance[i], strlen(names_dance[i]),1,fp2);
                i++;
                fflush(stdin);
        }
        fcloseall();
        fp1 = fopen(FILE1, "rb");
        if(fp1==NULL)
        {       printf("\n The file could not be opened");
                exit(1);
        }
        fp2 = fopen(FILE2, "rb");
        if(fp2==NULL)
        {       printf("\n The file could not be opened");
                exit(1);
        }
        i=0;
        while(feof(fp1)==0)
        {       fread(names_act[i],strlen(names_act[i]),1,fp1);
                i++;
        }
        n=i-1;
        fclose(fp1);
        for(i=0;i<n;i++)
        {       strcpy(names[j],names_act[i]);
                j++;
        }
        i=0;
        while(feof(fp2)==0)
        {       fread(names_dance[i],strlen(names_dance[i]),1,fp2);
```

Oxford University Press

Copyright© 2018 by Oxford University Press

```
                i++;
        }
        n=i-1;
        fclose(fp2);
        for(i=0;i<n;i++)
        {       strcpy(names[j],names_dance[i]);
                j++;
        }
        n=j;
        for(i=0;i<n;i++)
        {       for(j=0;j<n-i-1;j++)
                {       if(strcmp(names[j], names[j+1])>0)
                        {       strcpy(temp, names[j]);
                                strcpy(names[j], names[j+1]);
                                strcpy(names[j+1], temp);
                        }
                }
        }
        fp3 = fopen(FILE3, "wb");
        if(fp3==NULL)
        {       printf("\n The file could not be opened");
                exit(1);
        }
        for(i=0;i<n;i++)
                fwrite(names[i],strlen(names[i]),1,fp3);
}
```

21.

```
#include<stdio.h>
#include<stdlib.h>
int main()
{
FILE *fp1;
int num, sum=0;
fp1 = fopen("INTEGERS.TXT", "wb");
if(fp1==NULL)
{       printf("\n The file could not be opened");
        exit(1);
}
while(1)
{       printf("\n Enter a number : ");
        scanf("%d", &num);
        if(num==-1) break;
        sum +=num;
        fwrite(&num, sizeof(int),1,fp1);
}
fwrite(&sum, sizeof(int),1,fp1);
fclose(fp1);
}
```

23. You can use the files created in question 48 and then append one file at the end of other by reading records from the second file and writing them at the end of the other. The end of the first file can be reached by using fseek().

Find the Output of the Following Codes

Oxford University Press

Copyright© 2018 by Oxford University Press

1. 21 (depends on number of characters entered
2. Reads 5 comments from the user and then stores them in a file
3. C
4. A
5. G
6. 3
7. 11

Oxford University Press

Copyright© 2018 by Oxford University Press