**Question 01**

**Code**

#include<stdio.h>

#include<stdlib.h>

void main(void){

FILE \*f1=fopen("f1.txt","w");

if (f1==NULL){

printf("Error opening f1.txt\n");

exit(0);

}

fprintf(f1,"This is the text from f1\n");

fclose(f1);

FILE \*f2=fopen("f2.txt","w");

if (f2==NULL){

printf("Error opening f2.txt\n");

exit(0);

}

fprintf(f2,"This is the text from f2\n");

fclose(f2);

FILE \*input\_F1=fopen("f1.txt","r");

FILE \*input\_F2=fopen("f2.txt","r");

FILE \*output\_F=fopen("merged.txt","w");

if (input\_F1==NULL || input\_F2==NULL || output\_F==NULL){

printf("Error opening the files\n");

exit(0);

}

char ch;

while ((ch=fgetc(input\_F1))!=EOF){

fputc(ch,output\_F);

}

fprintf(output\_F,"\n");

while ((ch=fgetc(input\_F2))!= EOF){

fputc(ch,output\_F);

}

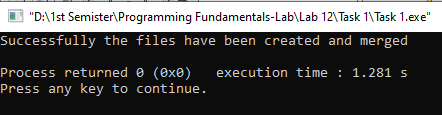
fclose(input\_F1);

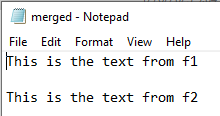
fclose(input\_F2);

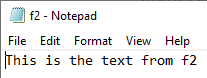
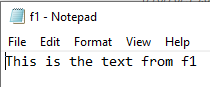
fclose(output\_F);

printf("Successfully the files have been created and merged\n");

}

**OUTPUT**





**Question 02**

**CODE**

#include<stdio.h>

#include<stdlib.h>

#include<ctype.h>

int count\_alpha(FILE \*ptr, int \*letters);

void main(void){

FILE \*f1;

FILE \*f2;

f1=fopen("text.txt","r");

if(f1==NULL){

printf("Error opening the file\n");

exit(0);

}

f2=fopen("output.txt","w");

if(f2==NULL){

printf("Error opening the file\n");

exit(0);

}

int letters[26]={0};

int total\_char;

int \*ptr=letters;

total\_char=count\_alpha(f1,ptr);

for(int i=0; i<26; i++){

fprintf(f2,"%c\t (Occurrences of %c / %c = %d)\n",'A'+i,'A'+i,'a'+i,ptr[i]);

}

fprintf(f2,"Total characters read=%d",total\_char);

fclose(f1);

fclose(f2);

}

int count\_alpha(FILE\* ptr, int\* letters){

char ch;

int total\_char=0;

ch=getc(ptr);

while(ch!=EOF){

if((ch>='a' && ch<='z') || (ch>='A' && ch<='Z')){

letters[tolower(ch)-'a']++;

total\_char++;

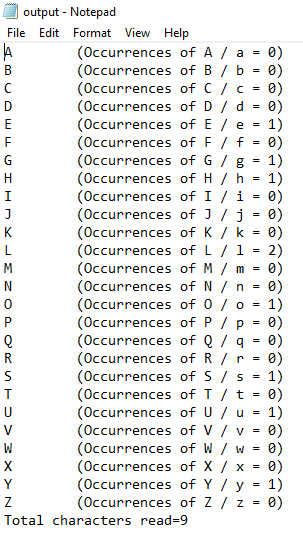
}

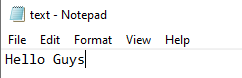
ch=getc(ptr);

}

return total\_char;

}

**OUTPUT**



**Question 03**

**CODE**

#include<stdio.h>

#include<string.h>

#include<stdlib.h>

struct student{

int id;

char name[30];

char sex;

float quiz\_1;

float quiz\_2;

float midterm;

float final\_mark;

float total;

};

void main(void){

FILE \*fptr=fopen("studentrecords.txt","w");

if (fptr==NULL){

printf("Error opening the file\n");

exit(0);

}

printf("Enter the number of students: ");

int n;

scanf("%d",&n);

struct student s[n];

int i;

for(i=0; i<n; i++){

printf("\nEnter the id of %d student:",i+1);

scanf("%d",&s[i].id);

printf("\nEnter the name of %d student: ",i+1);

fflush(stdin);

gets(s[i].name);

printf("\nSex (M=Male / F=Female): ");

fflush(stdin);

scanf("%c",&s[i].sex);

printf("\nEnter the score of QUIZ 1: ");

scanf("%f",&s[i].quiz\_1);

printf("\nEnter the score of QUIZ 2: ");

scanf("%f",&s[i].quiz\_2);

printf("Enter the score of Mid-term: ");

scanf("%f",&s[i].midterm);

printf("\nEnter the score of Final: ");

scanf("%f",&s[i].final\_mark);

s[i].total=s[i].quiz\_1+s[i].quiz\_2+s[i].midterm+s[i].final\_mark;

fprintf(fptr,"%d %s %c %.2f %.2f %.2f %.2f %.2f\n",s[i].id,s[i].name,s[i].sex,s[i].quiz\_1,s[i].quiz\_2,s[i].midterm,s[i].final\_mark,s[i].total);

}

fclose(fptr);

float score=0.0;

printf("\nEnter a score threshold (50, 80, 100) to analyze scores or 0 to end: ");

scanf("%f",&score);

while (score!=0){

int count\_below=0;

for(i=0; i<n; i++){

if(score==100 && s[i].total==100){

printf("Student %s scored 100 in total\n",s[i].name);

}else if(s[i].total<score){

count\_below++;

}

}

printf("Number of students with a total score below %.2f: %d\n",score,count\_below);

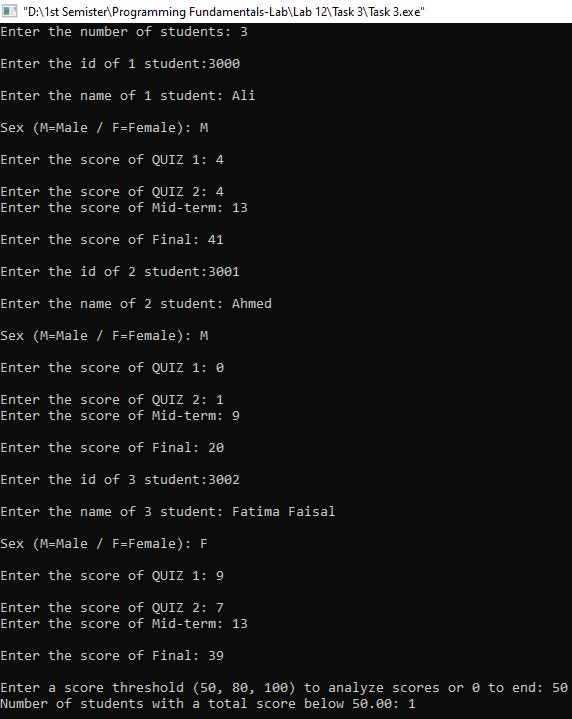
printf("\nEnter a score threshold (50, 80, 100) to analyze scores or 0 to end: ");

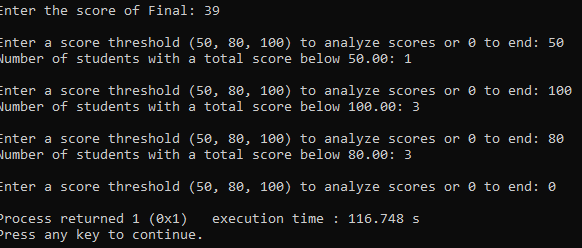
scanf("%f", &score);

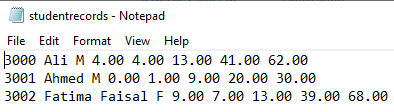
}

}

**OUTPUT**







**Question 04**

**CODE**

#include<stdio.h>

#include<string.h>

#include<stdlib.h>

struct tool{

int id;

char name[30];

int quantity;

float cost;

};

void initialize(){

FILE \*file=fopen("hardware.txt","w");

if(file==NULL){

printf("Error opening the file\n");

exit(0);

}

struct tool empty;

int i;

for (i=0; i<10; i++){

fwrite(&empty,sizeof(struct tool),1,file);

}

fclose(file);

}

void display(struct tool t){

printf("Tool ID: %d\n",t.id);

printf("Tool Name: %s\n",t.name);

printf("Quantity: %d\n",t.quantity);

printf("Cost: %.2f\n",t.cost);

printf("\n");

}

void list(){

FILE \*file=fopen("hardware.txt","r");

if (file==NULL){

printf("Error opening the file\n");

exit(0);

}

struct tool t;

printf("\n\tList of Tools:\n");

while (fread(&t, sizeof(struct tool),1, file) == 1) {

if (t.id!=0){

display(t);

}

}

fclose(file);

}

void add(){

FILE \*file=fopen("hardware.txt", "r+");

if (file==NULL){

printf("Error opening file for adding a tool.\n");

exit(1);

}

struct tool t;

int id;

printf("Enter the Tool ID: ");

scanf("%d",&id);

if (id<= 0|| id>10) {

printf("Invalid Tool ID\n");

fclose(file);

return;

}

fseek(file,(id-1)\*sizeof(struct tool),SEEK\_SET);

fread(&t,sizeof(struct tool),1,file);

if (t.id!=0){

printf("Tool with ID %d already exists therefore: use the update option to modify\n",id);

fclose(file);

return;

}

t.id=id;

printf("Enter the Tool Name: ");

fflush(stdin);

scanf("%s",t.name);

printf("Enter the quantity: ");

scanf("%d",&t.quantity);

printf("Enter the cost: ");

scanf("%f",&t.cost);

fseek(file,(id-1)\*sizeof(struct tool),SEEK\_SET);

fwrite(&t,sizeof(struct tool),1,file);

printf("Tool is successfully added\n");

fclose(file);

}

void delete\_(){

FILE \*file=fopen("hardware.txt","r+");

if(file==NULL){

printf("Error opening the file\n");

exit(0);

}

struct tool empty={0, "", 0, 0.0};

int id;

printf("Enter the Tool ID to delete: ");

scanf("%d",&id);

if (id<=0 || id>10){

printf("Invalid Tool ID\n");

fclose(file);

return;

}

fseek(file,(id-1)\*sizeof(struct tool),SEEK\_SET);

fwrite(&empty,sizeof(struct tool),1,file);

printf("Tool with %d ID is successfully deleted\n",id);

fclose(file);

}

void update(){

FILE \*file=fopen("hardware.txt","r+");

if(file==NULL){

printf("Error opening the file\n");

exit(0);

}

struct tool t;

int id;

printf("Enter the Tool ID you want to update: ");

scanf("%d",&id);

if(id<=0 || id>10){

printf("Invalid Tool ID\n");

fclose(file);

return;

}

fseek(file,(id-1)\*sizeof(struct tool),SEEK\_SET);

fread(&t,sizeof(struct tool),1,file);

if(t.id==0){

printf("Tool with ID %d does not exist therefore you can use the add option to create a new tool\n", id);

fclose(file);

return;

}

printf("Current details for Tool ID %d:\n",id);

display(t);

printf("Enter updated details for %d Tool ID:\n",id);

printf("Enter the Tool Name: ");

fflush(stdin);

scanf("%s",t.name);

printf("Enter the quantity: ");

scanf("%d",&t.quantity);

printf("Enter the cost: ");

scanf("%f",&t.cost);

fseek(file,(id-1)\*sizeof(struct tool),SEEK\_SET);

fwrite(&t,sizeof(struct tool),1,file);

printf("Tool with ID %d is successfully updated\n",id);

fclose(file);

}

void main(void){

initialize();

char choice;

do{

printf("\t\tPlease choose from the following options:\n");

printf("1) List all tools\n");

printf("2) Add a tool\n");

printf("3) Delete a tool\n");

printf("4) Update tool information\n");

printf("0) Exit\n");

printf("Enter your choice: ");

fflush(stdin);

choice=getchar();

switch(choice){

case '1':

list();

break;

case '2':

add();

break;

case '3':

delete\_();

break;

case '4':

update();

break;

case '0':

printf("Exiting the program\n");

break;

default:

printf("INVALID INPUT\n");

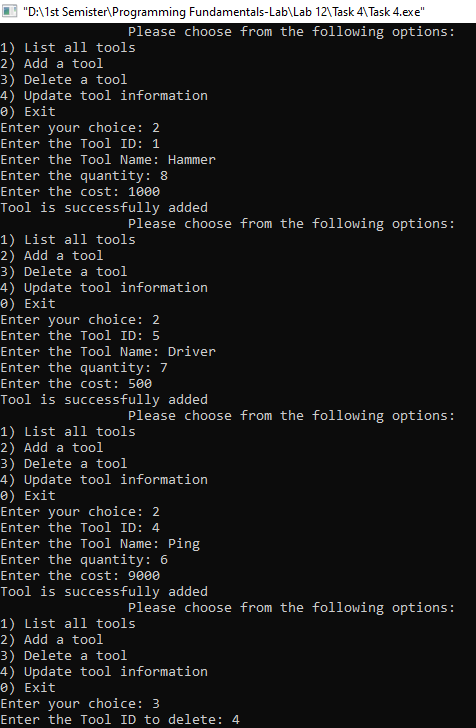
}

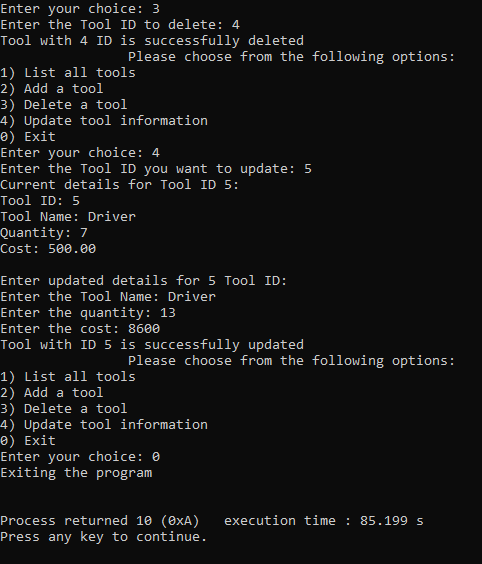
}while(choice!='0');

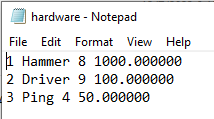
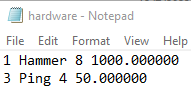
printf("\n");

}

**OUTPUT**







**Question 05**

**CODE**

#include<stdio.h>

#include<conio.h>

void main(void){

FILE \*fptr;

fptr=fopen("budget.txt","w");

if(fptr==NULL){

printf("Error opening the file\n");

exit(0);

}

fprintf(fptr,"-462.13 486.47 973.79\n");

fprintf(fptr," 755.42 843.04 -963.67\n");

fprintf(fptr," 442.5 -843.02 -462.86\n");

fprintf(fptr,"-233.93 -821.67 399.59\n");

fprintf(fptr,"-379.65 -556.37 837.46\n");

fprintf(fptr," 55.1 -144.93 -93.15\n");

fprintf(fptr," 533.73 804.64 -66.25\n");

fprintf(fptr,"-922.12 914.68 -264.67\n");

fprintf(fptr,"-600.27 -838.59 747.02\n");

fprintf(fptr,"-962.97 49.96 -677.79\n");

fclose(fptr);

fptr=fopen("budget.txt","r");

if(fptr==NULL){

printf("\nError opening the file\n");

exit(0);

}

float num,sum\_col1=0,sum\_col2=0,sum\_col3=0;

int i=0;

for(i=0;i<10;i++){

fscanf(fptr,"%f",&num);

sum\_col1=sum\_col1+num;

fscanf(fptr,"%f",&num);

sum\_col2=sum\_col2+num;

fscanf(fptr,"%f",&num);

sum\_col3=sum\_col3+num;

}

fclose(fptr);

printf("\n");

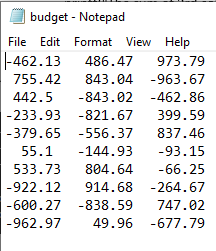
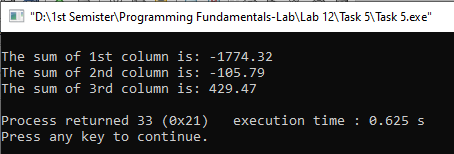
printf("The sum of 1st column is: %.2f\n",sum\_col1);

printf("The sum of 2nd column is: %.2f\n",sum\_col2);

printf("The sum of 3rd column is: %.2f\n",sum\_col3);

}

**OUTPUT**



**Question 06**

**CODE**

#include<stdio.h>

#include<string.h>

struct students {

int roll\_number;

char name[30];

char department[20];

int batch;

char section[10];

float cgpa;

};

void main(void){

int n;

printf("Please enter the number of students data you want to store: ");

scanf("%d",&n);

struct students s[n];

int i;

for (i=0; i<n; i++) {

printf("\nEnter the roll number of %d student: ",i+1);

scanf("%d",&s[i].roll\_number);

printf("\nEnter the name of %d student: ",i+1);

fflush(stdin);

gets(s[i].name);

printf("\nEnter the department of %d student: ",i+1);

fflush(stdin);

gets(s[i].department);

printf("\nEnter the batch number of %d student: ",i+1);

scanf("%d",&s[i].batch);

printf("\nEnter the section of %d student: ",i+1);

fflush(stdin);

gets(s[i].section);

printf("\nEnter the CGPA of %d student: ",i+1);

scanf("%f",&s[i].cgpa);

printf("\n");

}

FILE \*file=fopen("file.txt","w");

if(file==NULL){

printf("\nError opening the file\n");

exit(0);

}

for(i=0; i<n; i++){

fprintf(file,"%d %s %s %d %s %f\n",s[i].roll\_number,s[i].name,s[i].department,s[i].batch,s[i].section,s[i].cgpa);

}

fclose(file);

file=fopen("file.txt","r");

if(file==NULL){

printf("\nError opening the file\n");

exit(0);

}

int search;

printf("Enter the roll number you want to search for: ");

scanf("%d",&search);

int index=0,found=0;

while(fscanf(file,"%d %s %s %d %s %f",&s[index].roll\_number,s[index].name,s[index].department,&s[index].batch,s[index].section,&s[index].cgpa)!=EOF){

if(search==s[index].roll\_number){

printf("\nThe roll number of student is: ");

printf("%d",s[index].roll\_number);

printf("\nThe name of student is: ");

puts(s[index].name);

printf("\nThe department of student is: ");

puts(s[index].department);

printf("\nThe batch number of student is: ");

printf("%d",s[index].batch);

printf("\nThe section of student is: ");

puts(s[index].section);

printf("\nCGPA of student is: ");

printf("%f",s[index].cgpa);

printf("\n");

found=1;

break;

}

index++;

}

fclose(file);

file=fopen("file.txt", "r");

if(file==NULL){

printf("\nError opening the file\n");

exit(0);

}

int Batch=2022,read=0;

printf("\nStudents in Batch %d: \n",Batch);

rewind(file);

while(fscanf(file,"%d %s %s %d %s %f",&s[read].roll\_number,s[read].name,s[read].department,&s[read].batch,s[read].section,&s[read].cgpa)!=EOF){

if(s[read].batch==Batch){

printf("\nThe roll number of student is: ");

printf("%d",s[read].roll\_number);

printf("\nThe name of student is: ");

puts(s[read].name);

printf("\nThe department of student is: ");

puts(s[read].department);

printf("The batch number of student is: ");

printf("%d",s[read].batch);

printf("\nThe section of student is: ");

puts(s[read].section);

printf("\nCGPA of student is: ");

printf("%f",s[read].cgpa);

printf("\n");

}

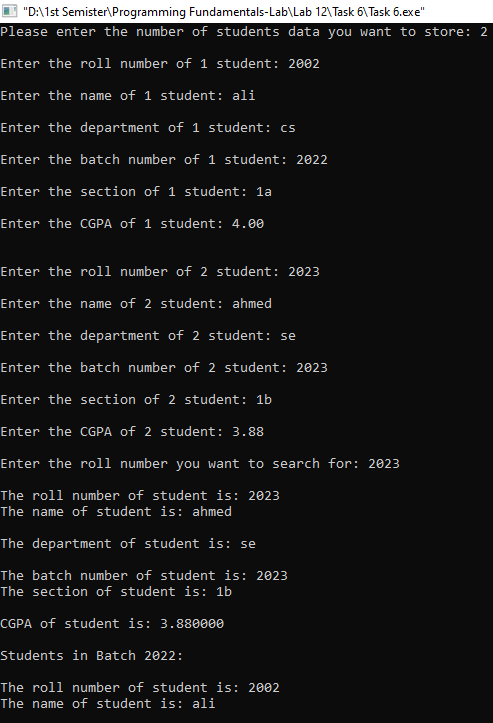
read++;

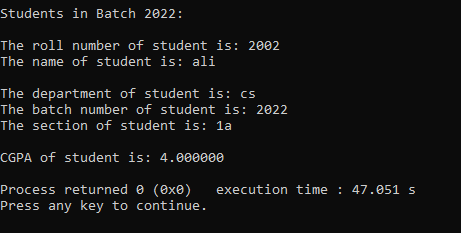
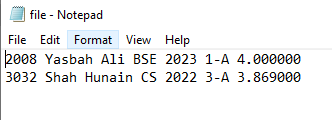
}

fclose(file);

}

**OUTPUT**





**Question 07**

**CODE**

#include<stdio.h>

#include<ctype.h>

#include<stdlib.h>

int vowel(char c){

return (c=='a' || c=='e' || c=='i' || c=='o' || c=='u' || c== 'A' || c=='E' || c=='I' || c== 'O' || c=='U');

}

char encrypt\_data(char c){

if (isalpha(c)){

char base;

if(islower(c)){

base='a';

}

else{

base='A';

}

return(c-base+3)%26+base;

}

else{

return c;

}

}

void main(void){

char c;

int count\_vowels = 0;

FILE \*f1=fopen("input.txt", "r");

if (f1==NULL){

printf("Error opening file input.txt\n");

exit(0);

}

FILE \*f2=fopen("output.txt", "w");

if (f2==NULL){

printf("Error opening the file\n");

exit(0);

}

while((c=fgetc(f1))!= EOF){

if (vowel(c)){

count\_vowels++;

if (count\_vowels%2==1){

fprintf(f2,"vow");

}

else{

fprintf(f2,"VOW");

}

}

else if(c=='s'||c=='S'){

char buffer[6]={'P','F','-','L','a','b'};

fwrite(buffer,sizeof(char), 4, f2);

}

else{

fprintf(f2,"%c",encrypt\_data(c));

}

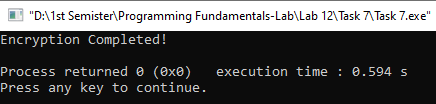
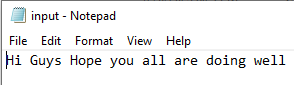
}

fclose(f1);

fclose(f2);

printf("Encryption Completed!\n");

}

**OUTPUT**

