

Introduction

14 July 2020 16:03

Source :- <https://github.com/hiverkiya/Let-Us-C-Solutions>

Chapter-1 : Getting Started

[A]

REASON:- A character constant is a constant which uses single quotation around characters.

Invalid:- '3.15' eLearning' 'Quest'

REASON:- No Comma/Space in integer constant & No exponentiation operator available.

Invalid:- "35,550" "2^3" "4 6 5 2"

[B]

REASON:- No special symbol except underscore allowed in variable naming

B'day \$hello #HASH _main() total% %name% stack-queue

REASON:- Keyword cannot be used as a variable.

int

REASON:- Variable name should start with an alphabet.

Same as 1st

[C]

- (a) True
- (b) True
- (c) True
- (d) True
- (e) True
- (f) True
- (g) True
- (h) True
- (i) True
- (j) False
- (k) True
- (l) False

[D]

- (a) Escape Sequence
- (b) Real Constant
- (c) Integer Constant
- (d) Character Constant
- (e) Exponential Form
- (f) Function
- (g) Format Specifier
- (h) Statement Terminator
- (i) Literal
- (j) Identifier
- (k) Address of Operator
- (l) Output Function

(m) Input Function

[E]

- (a) Statement Terminator is missing, and it couldn't be figured out whether the expression is $(a+b)*(b-35)$ or $a+(b*b)-35$ because of unavailability of parentheses.
- (b) Nesting of comments is not allowed.
- (c) "&" Operator is Missing.
- (d) Statement terminator operator ";" is missing in integer declaration and scanf & printf functions.

[F]

(a)

```
#include<stdio.h>
int main()
{
    float bs,gs;
    printf("Enter basic salary of Rakesh.\n");
    scanf("%f",&bs);
    printf("Gross salary of Rakesh is %f.\n", (bs+(0.4*bs)+(0.2*bs)));
    return 0;
}
```

OR

```
#include<stdio.h>
#include<conio.h>
int main()
{
    int sal;
    float gros_sal;
    printf("Enter the basic salary of Ramesh : ");
    scanf("%d", &sal);
    //salary calculated
    gros_sal=sal+(sal*40/100)+(sal*20/100);
    printf("%f is the gross salary of Ramesh.", gros_sal);
    getch();
    return 0;
}
```

(b)

```
#include<stdio.h>
int main()
{
    float meters,feet,inches,centi_m,km;
    printf("Enter the distance in km\n");
    scanf("%f",&km);
    meters=km*1000;
    centi_m=meters*100;
    feet=3.28084*meters;
    inches=12*feet;
    printf("The distance in meters = %f\n",meters);
    printf("The distance in centimeters = %f\n",centi_m);
    printf("The distance in feet = %f\n",feet);
    printf("The distance in inches = %f\n",inches);
    return 0;
}
```

OR

```
#include<stdio.h>
#include<conio.h>
int main()
{
    float dis;
    printf("Enter the distance between cities in kilo metres : ");
    scanf("%f", &dis);
    printf("\n%f is the distance between them in meters.", dis*1000);
    printf("\n%f is the distance between them in feet.", dis*3280.8399);
    printf("\n%f is the distance between them in inches.", dis*39370.0788);
    printf("\n%f is the distnace between them in centimeters", dis*100000);
    getch();
    return 0;
}
```

(c)

```
#include<stdio.h>
int main()
{
    printf("Enter marks of student\n");
    float array[5],aggregate=0,percentage;
    int i;
    for(i=0;i<5;i++)
    {
        scanf("%f",&array[i]);
        aggregate+=array[i];
    }
    printf("Aggregate Marks of Student are :- %f.\n",aggregate);
    percentage=aggregate/500*100;
    printf("Percentage of Students is %f.\n",percentage);
    return 0;
}
```

OR

```
#include<stdio.h>
#include<conio.h>
int main()
{
    float a,b,c,d,e;
    printf("Enter the marks out of 100 in five subjects of the student : ");
    scanf("%f%f%f%f%f", &a,&b,&c,&d,&e);
    //his total marks
    printf("\n%f is the aggregate marks obtained by him.", a+b+c+d+e);
    //his percentage marks
    printf("\n%f is the percentage marks obtained by him.", (a+b+c+d+e)/5);
    getch();
    return 0;
}
```

(d)

```
#include<stdio.h>
int main()
{
    float temp_in_c,temp_in_f;
    printf("Enter the temperature in Fahrenheit\n");
    scanf("%f",&temp_in_f);
    printf("Temperature in Celsius = %f.",(temp_in_f-32)/1.8);
    return 0;
}
```

OR

```
#include<stdio.h>
#include<conio.h>
int main()
{
    float f,c;
    printf("Enter the temperature in Fahrenheit : ");
    scanf("%f", &f);
    //conversion of temperature
    c=(f-32)*100/180;
    printf("\n%f is that temperature in celcius.", c);
    getch();
    return 0;
}
```

(e)

```
#define PI 3.14159265
#include<stdio.h>
int main()
{
    printf("Enter the length and breadth of rectangle\n");
    float length,breadth;
    scanf("%f %f",&length,&breadth);
    printf("Enter the radius of circle\n");
    float radius,area=0;
    scanf("%f",&radius);
    printf("Area of Rectangle is %f and Perimeter is %f.\n",length*breadth,2*(length+breadth));
}
```

```

printf("Area of Circle is %f and Circumference is %f.\n",PI*radius*radius,2*PI*radius);
return 0;
}

```

OR

```

#include<stdio.h>
#include<conio.h>
int main()
{
    float l,b,r;
    printf("Enter the length and breadth of the rectangle : ");
    scanf("%f%f", &l,&b);
    //rectangle calculations
    printf("\n%f is the area of the rectangle.", l*b);
    printf("\n%f is the perimeter of the rectangle.", l+b);
    //circle calculations
    printf("\n\nEnter the radius of the circle : ");
    scanf("%f", &r);
    printf("\n%f is the area of the circle.", 3.14*r*r);
    printf("\n%f is the perimeter of the circle.", 2*3.14*r);
    getch();
    return 0;
}

```

(f)

```

#include <stdio.h>
int main()
{
    int length=1189,breadth=841,temp,i;
    for(i=0;i<=8;i++)
    {
        printf("The size of A(%d) sheet = %d mm x %d mm.\n",i,length,breadth);
        temp=length;
        length=breadth;
        breadth=temp/2;
    }
    return 0;
}

```

OR

```

#include<stdio.h>
#include<conio.h>
int main()
{
    int A0_a, A0_b, A1_a, A1_b, A2_a, A2_b, A3_a, A3_b, A4_a, A4_b, A5_a, A5_b, A6_a, A6_b, A7_a, A7_b, A8_a, A8_b;
    A0_a = 1189;
    A0_b = 841;
    printf("\nA0 : %dmm x %dmm ", A0_a, A0_b);
    A1_a = A0_b;
    A1_b = A0_a/2;
    printf("\nA1 : %dmm x %dmm ", A1_a, A1_b);
    A2_a = A1_b;
    A2_b = A1_a/2;
    printf("\nA2 : %dmm x %dmm", A2_a, A2_b);
    A3_a = A2_b;
    A3_b = A2_a/2;
    printf("\nA3 : %dmm x %dmm", A3_a, A3_b);
    A4_a = A3_b;
    A4_b = A3_a/2;
    printf("\nA4 : %dmm x %dmm", A4_a, A4_b);
    A5_a = A4_b;
    A5_b = A4_a/2;
    printf("\nA5 : %dmm x %dmm", A5_a, A5_b);
    A6_a = A5_b;
    A6_b = A5_a/2;
    printf("\nA6 : %dmm x %dmm", A6_a, A6_b);
    A7_a = A6_b;
    A7_b = A6_a/2;
    printf("\nA7 : %dmm x %dmm", A7_a, A7_b);
    A8_a = A7_b;
    A8_b = A7_a/2;
    printf("\nA8 : %dmm x %dmm", A8_a, A8_b);
    getch();
    return 0;
}

```


Instructions

14 July 2020 16:06

<https://github.com/hiverkiya/Let-Us-C-Solutions>

Chapter-2 Instructions

[A]

- (a) Valid.
- (b) Valid.
- (c) "char" is a keyword and we cannot use keyword as a variable.
- (d) lvalue required, as we cannot take any expression on LHS.
- (e) a3 doesn't specify any operation.
- (f) Valid.
- (g) Valid.
- (h) spaces are not allowed in variable names.
- (i) Valid.
- (j) "?"(symbol) is not any valid operator.
- (k) lvalue required as LHS should not have any constant value or any expression.
- (l) Valid.
- (m) Length of character is one.

[B]

- (a) -84
- (b) 4
- (c) INFINITE
- (d) 23.750000

[C]

Operators are performed from LHS in the given expressions.

- (a) / / /
- (b) / + * /
- (c) = = = +
- (d) = - + * / %
- (e) = % / * +
- (f) = = % / +

[D]

- (a) $(x + 3) * x * x * x / ((y - 4) * (y + 5));$
- (b) $2 * v + 6.22 * (c + d) / (g + v);$

(c) $7.7 * b (x * y + a) / c - 0.8 + 2 * b / ((x + a)*(1 / y));$

(d) $(12 * x * x * x * x) / (4 * x) + (8 * x * x) / (4 * x) + (x / (8 * x)) + 8 / (8 * x);$

[E]

(a) 0 2 0.000000 2.000000

(b) a=2 b=-2 c=2 d=-2

(c) 2

(d) Compile error

(e) Respective values of a & b

[F]

(a) True

(b) False

(c) True

(d) True

(e) False

(f) True

[G]

(a) $10*x$

(b) 5

(c) 5

(d) -9

(e) 0.285714

[H]

(a)

```
#include<stdio.h>
int main()
{
    int number,sum=0;
    printf("Enter number\n");
    scanf("%d",&number);
    while(number!=0)
    {
        sum+=number%10;
        number/=10;
    }
    printf("Sum of digits is %d",sum);
    return 0;
}
```

OR

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

```

int num,a,b,c,d,e;
printf("Enter a five digit number : ");
scanf("%d", &num);
e=num%10;
d=(num/10)%10;
c=(num/100)%10;
b=(num/1000)%10;
a=(num/10000);
printf("%d is the sum of the digits of the number %d.", a+b+c+d+e, num);
getch();
return 0;
}

```

(b)

```

#include<stdio.h>
#include<math.h>
int main()
{
printf("Enter number\n");
int number;
scanf("%d",&number);
int temp=number;
int length_of_number=5;
int reverse=0,remainder=0;
while(number!=0)
{
length_of_number--;
remainder=number%10;
reverse=reverse+remainder*pow(10.0,length_of_number);
number/=10;
}
printf("Reverse of number %d is %d",temp,reverse);
return 0;
}

```

OR

```

#include<stdio.h>
#include<conio.h>
int main()
{
int num,a,b,c,d,e;
printf("Enter a five digit number : ");
scanf("%d", &num);
e=num%10;
d=(num/10)%10;
c=(num/100)%10;
b=(num/1000)%10;
a=(num/10000);
num = e*10000+d*1000+c*100+b*10+a;
printf("\n%d", num);
getch();
return 0;
}

```

(c)

```

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

```



```

float side1,side2,side3,s_p;
printf("Enter sides of triangle\n");
scanf("%f %f %f",&side1,&side2,&side3);
s_p=(side3+side2+side1)/2.0;
printf("Area of triangle is %f",sqrt(s_p*(s_p-side1)*(s_p-side2)*(s_p-
side3)));
return 0;
}

```

OR

```

#include<stdio.h>
#include<conio.h>
#include<math.h>
int main()
{
    float a , b , c , s , area;

    printf("\nEnter three sides of the triangle : ");
    scanf("%f%f%f", &a, &b, &c);

    //Finding area of the triangle by the Heron's Formula
    s = (a+b+c)/2;
    area = sqrt(s*(s-a)*(s-b)*(s-c));
    //sqrt() is a funtion that find the square root.

    printf("\nArea : %f", area);

    getch();
    return 0;
}

```

(d)

```

#include<stdio.h>
#include<math.h>
int main()
{
    printf("Enter x and y coordinates \n");
    float x,y;
    scanf("%f %f",&x,&y);
    printf("Polar coordinates are (%f,%f).
\n",sqrt(pow(x,2.0)+pow(y,2.0)),atan(y/x));
    return 0;
}

```

OR

```

#include<stdio.h>>
#include<conio.h>
#include<math.h>
int main()
{
    float x, y, r, theta;

    printf("Enter the x and y coordinates : ");
    scanf("%f%f", &x, &y);

    r = sqrt(x*x + y*y);
    theta = atan(y/x);
    //atan() is a function to find the tan inverse

    printf("\nCoordinates in polar form : %.2f(cos(%.2f) + i.sin(%.2f))", r, t

```

```

heta, theta);
    //%.2f means only two decimal palces will be printed.

    getch();
    return 0;
}
(e)

```

```

#include<stdio.h>
#include<math.h>
int main()
{
    float l1,l2,g1,g2;
    printf("Enter values of latitude (l1,l2)\n");
    scanf("%f %f",&l1,&l2);
    printf("Enter values of longitude(g1,g2) in degrees\n");
    scanf("%f %f",&g1,&g2);
    printf("The Distance between them is %f\n",(3963
*acos(sin(l1)*sin(l2)+cos(l1)*cos(l2)*cos(g2-g1))));
    return 0;
}

```

OR

```

#include<stdio.h>
#include<conio.h>
#include<math.h>
int main()
{
    float l1,l2,g1,g2, D;

    printf("\nEnter (two) the values of lattitude : ");
    scanf("%f%f", &l1, &l2);

    printf("\nEnter (two) the values of longitude : ");
    scanf("%f%f", &g1, &g2);

    D = 3963*acos(sin(l1)*sin(l2) + cos(l1)*cos(l2)*cos(g2-g1));

    printf("\nDistance : %f", D);

    getch();
    return 0;
}

```

(f)

```

#include<math.h>
#include<stdio.h>
int main()
{
    printf("Enter temperature and velocity of wind\n");
    float temp,vel,wcf=0;
    scanf("%f %f",&temp,&vel);
    wcf=35.74+0.6215*temp+(0.4275*temp-35.75)*pow(vel,0.16);
    printf("Wind Chill factor of Temperature %f and Velocity %f is %
f",temp,vel,wcf);
    return 0;
}

```

OR

```

#include<stdio.h>
#include<conio.h>
#include<math.h>
int main()
{
    float t, v, wcf;

```

```
printf("\nEnter the values of temperature and wind velocity: ");
scanf("%f%f", &t, &v);
```

```
wcf = 35.74 + 0.6215*t + (0.42751*t - 35.75)*pow(v, 0.16);
//pow(a,b) function is used to calculate a^b
```

```
printf("Wind Chill Factor : %f", wcf);
```

```
getch();
return 0;
}
```

(g)

```
#include<stdio.h>
#include<math.h>
int main()
{
printf("Enter the value of angle\n");
float angle;
scanf("%f",&angle);
printf("The sine of angle = %f.\n",sin(angle));
printf("The cosine of angle = %f.\n",cos(angle));
printf("The tangent of angle = %f.\n",tan(angle));
printf("The cotangent of angle = %f.\n",(1/tan(angle)));
printf("The cosecant of angle = %f.\n",(1/sin(angle)));
printf("The secant of angle = %f.\n",(1/cos(angle)));
return 0;
}
```

(h)

```
#include<stdio.h>
int main()
{
printf("Enter two numbers\n");
int c,d;
scanf("%d %d",&c,&d);
printf("Before Interchange C=%d and D=%d.\n",c,d);
c=c+d;
d=c-d;
c=c-d;
printf("After Interchange C=%d and D=%d.\n",c,d);
return 0;
}
```

(i)

```
#include<stdio.h>
#include<conio.h>
int main()
{
int n100, n50, n10, n5, n2, n1, num;

printf("\nEnter the amount : ");
scanf("%d", &num);

//Numbers of notes of 100
n100 = num / 100;
num = num % 100;

//Numbers of notes of 50
n50 = num / 50;
num = num % 50;

//Numbers of notes of 10
```

```
n10 = num / 10;
num = num % 10;

//Numbers of notes of 5
n5 = num / 5;
num = num % 5;

//Numbers of notes of 2
n2 = num / 2;
num = num % 2;

//Numbers of notes of 1
n1 = num / 1;

printf("\n\nTo give amount of %d you have to give : \n");
printf("%d of notes of hundred.\n", n100);
printf("%d of notes of fifty.\n", n50);
printf("%d of notes of ten.\n", n10);
printf("%d of notes of five.\n", n5);
printf("%d of notes of two.\n", n2);
printf("%d of notes of one.\n", n1);

getch();
return 0;
}
```

Decision Control Instructions

14 July 2020 16:06

<https://github.com/hiverkiya/Let-Us-C-Solutions>

[Chapter 3: Decision Control Instruction] Solutions

[A]

- (a) garbage 200
- (b) 300 200
- (c) no output
- (d) x and y are equal
- (e) x=10 y=10 z=0
- (f) C is WOW

[B]

- (a) NO ERROR.
- (b) NO ERROR.
- (c) NO ERROR.
- (d) There is no keyword "then" related to if in C & no variable defined here in program as well.
- (e) lvalue required as left operand of assignment.
- (f) There should be a space between else and if, as there is no "elseif" syntax in C, "else if" exists.
- (g) "&" Operator is missing before variables "a" and "b".

[C]

(a)

```
#include<stdio.h>
int main()
{
    float sp,cp;
    printf("Enter Cost Price and Selling Price\n");
    scanf("%f %f",&cp,&sp);
    if(sp-cp>0)
    {
        printf("Profit incurred is %f.",sp-cp);
    }
    if(sp-cp<0)
    {
        printf("Loss incurred is %f",cp-sp);
    }
    return 0;
}
```

(b)

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

```

printf("Enter number\n");
int number;
scanf("%d",&number);
if(number%2==0)
{
printf("Number is Even");
}
else
{
printf("Number is Odd");
}
return 0;
}

```

OR

```

#include<stdio.h>;
#include<conio.h>;
int main()
{
int num;
printf("Enter an integer : ");
scanf("%d", &num);
if(num%2 != 0)//odd
printf("\n\nThe number is odd");
else//even
printf("\n\nThe number is even.");
getch();
return 0;
}

```

(c)

```

#include <stdio.h>
int main()
{
    int year;
    printf("Enter a year: ");
    scanf("%d",&year);
    if(year%4 == 0)
    {
        if( year%100 == 0)
        {
            // year is divisible by 400, hence the year is a leap year
            if ( year%400 == 0)
                printf("%d is a leap year.", year);
            else
                printf("%d is not a leap year.", year);
        }
        else
            printf("%d is a leap year.", year );
    }
    else
        printf("%d is not a leap year.", year);

    return 0;
}

```

(d)

```

#include<stdio.h>
int main()
{
int leapdays,firstday,yr;
long int normaldays,totaldays;
printf("enter the year\n");

```

```

scanf("%d",&yr);
normaldays= (yr-1)*365L;
leapdays=(yr-1)/4- (yr-1)/100+(yr-1)/400;
totaldays=normaldays+leapdays;
firstday=totaldays%7;
if(firstday==0)
printf("monday");
if(firstday==1)
printf("tuesday");
if(firstday==2)
printf("wednesday");
if(firstday==3)
printf("thursday");
if(firstday==4)
printf("friday");
if(firstday==5)
printf("saturday");
if(firstday==6)
printf("sunday");
return 0;
}

```

(e)

```

#include<stdio.h>
#include<math.h>
int main()
{
    printf("Enter number\n");
    int number;
    scanf("%d",&number);
    int temp=number;
    int length_of_number=5;
    int reverse=0,remainder=0;
    while(number!=0)
    {
        length_of_number--;
        remainder=number%10;
        reverse=reverse+remainder*pow(10.0,length_of_number);
        number/=10;
    }
    printf("Reverse of number %d is %d\n",temp,reverse);
    if(temp==reverse)
    {
        printf("Reverse and Original Number are Equal\n");
    }
    else
    {
        printf("Reverse and Original Number are not Equal");
    }
    return 0;
}

```

OR

```

#include<stdio.h>
#include<conio.h>
int main()
{
    int num,a,b,c,d,e,x;

    printf("Enter a five digit number : ");
    scanf("%d", &num);

```

```

//separating digits of the number
e = num % 10;
d = (num/10) % 10;
c = (num/100) % 10;
b = (num/1000) % 10;
a = (num/10000);

//reversing the number
x = e*10000 + d*1000 + c*100 + b*10 + a;
printf("\n%d", x);

if(x == num)
    printf("\n\nThe reverse of the number %
d is same as actual number.", num);

getch();
return 0;
}

(f)
#include<math.h>
#include<stdio.h>
int main()
{
    printf("Enter ages of Ram,Shyam and Ajay.\n");
    int age1,age2,age3;
    scanf("%d %d %d",&age1,&age2,&age3);
    int youngest;
    youngest=age1<age2?(age1<age3?age1:age3):(age2<age3?age2:age3);
    if(youngest==age1)
    {
        printf("Ram is Youngest.");
    }
    else if(youngest==age2)
    {
        printf("Shyam is Youngest.");
    }
    else if(youngest==age3)
    {
        printf("Ajay is Youngest.");
    }
    return 0;
}

```

OR

```

#include<stdio.h>
#include<conio.h>
int main()
{
    int ram,sam,ajy;

    printf("Enter the age of Ram, Shyam and Ajay respectively : ");
    scanf("%d%d%d", &ram,&sam,&ajy);

    if(ram<sam && ram<ajy)
        printf("\n\nRam is youngest among all.");
    else if(sam<ram && sam<ajy)
        printf("\n\nShyam is youngest among all.");
    else
        printf("\n\nAjay is youngest among all.");
}

```



```

    getch();
    return 0;
}
(g)
#include<stdio.h>
int main()
{
    int angle1,angle2,angle3;
    printf("Enter the angles of Triangle in any order in Degrees.\n");
    scanf("%d %d %d",&angle1,&angle2,&angle3);
    if(angle1+angle2+angle3==180)
    {
        printf("Triangle is Valid.");
    }
    else
    {
        printf("Triangle is Invalid.");
    }
    return 0;
}
(h)
#include<math.h>
#include<stdio.h>
int main()
{
    printf("Enter number\n");
    int number;
    scanf("%d",&number);
    printf("Absolute value of %d is %d.",number,abs(number));
    return 0;
}

```

OR

```

#include<stdio.h>
#include<conio.h>
int main()
{
    int a;

    printf("Enter any number : ");
    scanf("%d", &a);

    if(a>0)
        printf("\n\n%d is it's absolute value.", a);
    else
        printf("\n\n%d is it's absolute number.", -1*a);

    getch();
    return 0;
}
(i)
#include <stdio.h>
int main()
{
    int l,b;
    printf("Enter length and breadth of rectangle\n");
    scanf("%d %d",&l,&b);
    if((l*b)>(2*(l+b)))
    {
        printf("Area is Greater than perimeter\n");
    }
}

```

```

else
{
    printf("Area is not Greater than perimeter\n");
}
return 0;
}

```

OR

```

#include<stdio.h>
#include<conio.h>
int main()
{
    float a,b, area, peri;

    printf("Enter the length and breadth of the rectangle : ");
    scanf("%f%f", &a,&b);

    area = a*b;
    peri = 2*a + 2*b;

    if((area > peri)
        printf("\n\nThe area %
f of the rectangle is greater than it's perimenter %f.", area,peri);
    else
        printf("\n\nThe area %
f of the rectangle is less than it's perimenter %f.", area,peri);

    getch();
    return 0;
}

```

(j)

```

#include <stdio.h>
int main()
{
    float x1,y1,x2,y2,x3,y3;
    printf("Enter coordinates (x1,y1)\n");
    scanf("%f %f",&x1,&y1);
    printf("Enter coordinates (x2,y2)\n");
    scanf("%f %f",&x2,&y2);
    printf("Enter coordinates (x3,y3)\n");
    scanf("%f %f",&x3,&y3);
    if((y2-y1)/(x2-x1)==(y3-y1)/(x3-x1))
    {
        printf("Point lies on straight line");
    }
    else
    {
        printf("Points don't lie on straight line");
    }
    return 0;
}

```

OR

```

#include<stdio.h>
#include<conio.h>
int main()
{
    int x1,x2,x3,y1,y2,y3,ar;

```

```

printf("Enter x-y coordinates of first point : ");
scanf("%f%f", &x1,&y1);

printf("\nEnter x-y coordinates of second point : ");
scanf("%f%f", &x2,&y2);

printf("\nEnter x-y coordinates of third point : ");
scanf("%f%f", &x3,&y3);

ar= (x1*(y2-y3) + x2*(y3-y1) + x3*(y1-y2)); //condition for collinear

if(!ar)
    printf("\n\nThe points are collinear.");
else
    printf("\n\nThe points are not collinear.");

//To check there's is an example (0, -2) , (2, 4) and (-1, -5).

getch();
return 0;
}

```

(k)

```

#include<stdio.h>
int main()
{
    float cen_x,cen_y,rad,x,y,d;
    printf("Enter Center x,y coodinate and radius\n");
    scanf("%f %f %f",&cen_x,&cen_y,&rad);
    printf("Enter coordinates of point\n");
    scanf("%f %f",&x,&y);
    d=sqrt(pow((cen_x-x),2.0)+pow((cen_y-y),2.0));
    if(d<rad)
    {
        printf("Point is inside circle\n");
    }
    else if(d==rad)
    {
        printf("Point is on the circle\n");
    }
    else if(d>rad)
    {
        printf("Point is outside the circle\n");
    }
    return 0;
}

```

OR

```

#include<stdio.h>
#include<conio.h>
int main()
{
    float r,x,y;

    printf("Enter the radius of the circle : ");
    scanf("%f", &r);

    printf("\n\nEnter the x-
y coordinates of the point for checking it's position : ");
    scanf("%f%f", &x,&y);

    if(r > sqrt(pow(x,2) + pow(y,2)))

```

```

        printf("\n\nThe points lie inside the circle.");
    else if(r == sqrt(pow(x,2) + pow(y,2)))
        printf("\n\nThe points lie on the circle.");
    else
        printf("\n\nThe points lie outside the circle.");

    getch();
    return 0;
}
(1)
#include<stdio.h>
int main()
{
    float x,y;
    printf("Enter Coordinates\n");
    scanf("%f %f",&x,&y);
    if(x==0.0&&y==0.0)
    {
        printf("Point lies on origin\n");
    }
    else if (x==0.0&&y>0.0)
    {
        printf("Point lies on y-axis\n");
    }
    else if(y==0.0&& x>0.0)
    {
        printf("Point lies on x-axis\n");
    }
    return 0;
}

```

OR

```

#include<stdio.h>
#include<conio.h>
int main()
{
    float x,y;

    printf("Enter the x-y coordinates of the point : ");
    scanf("%f%f", &x,&y);

    if(x==0 && y==0)
        printf("\n\nThe point is on the origin.");
    if(x==0 && y!=0)
        printf("\n\nThe point lie on the y-axis");
    if(x!=0 && y==0)
        printf("\n\nThe points lie on the x-axis");
    if(x!=0 && y!=0)
        printf("\n\nThe points lie on the plane");

    getch();
    return 0;
}

```

More Complex Decision Making

14 July 2020 16:07

Chapter-4 More Complex Decision Making

[A]

```
Value: 1
      0

      1

      1

      1
```

[B]

- (a) Dean of students affairs
- (b) w=1 x=0 y=1 z=1
- (c) biggest=45
- (d) -1 1
- (e) 1
- (f) Compile error

[C]

- (a) NO ERROR.
- (b) NO ERROR.
- (c) Wrong use of "&&" Operators.
- (d) NO ERROR.
- (e) "else" used without previous "if" as first "if" is terminated with ";"
- (f) NO ERROR.
- (g) NO ERROR. However, result will be j>=65?9:
- (h) lvalue required as left operand of assignment
- (i) Wrong use of ternary operator "? then :".
- (j) There is no use of ";" after printf().

[D]

```
(a)
#include<stdio.h>
int main()
{
    int year;
    scanf("%d",&year);
    if(year%4==0 || year%100==0 || year%400==0)
        printf("%d is a leap year",year);
    else
        printf("%d is not a leap year",year);
    return 0;
}
```

(b)

```
#include<stdio.h>
int main()
{
    char character;
    scanf("%c",&character);
    if(character>=65 && character<=90)
        printf("%c is a capital letter",character);
    else if(character>=97 && character<=122)
        printf("%c is a small letter",character);
    else if(character>=48 && character<=57)
        printf("%c is a digit",character);
    else
        printf("%c is a special symbol",character);
    return 0;
}
```

(c)

```
#include<stdio.h>
int main()
{
    int hard,tensile;
    float carbon;
    scanf("%d",&hard);
    scanf("%f",&carbon);
    scanf("%d",&tensile);
    if(hard>50 && carbon<0.7 && tensile>5600)
        printf("Grade is 10");
    else if(hard>50 && carbon<0.7 && tensile<=5600)
        printf("Grade is 9");
    else if(hard<=50 && carbon<0.7 && tensile>5600)
        printf("Grade is 8");
    else if(hard>50 && carbon>=0.7 && tensile>5600)
        printf("Grade is 7");
    else
        if(hard>50 && carbon>=0.7 && tensile<=5600 || hard<=50 && carbon<0.7
        && tensile<=5600 || hard<=50 && carbon>=0.7 && tensile>5600)
            printf("Grade is 6");
        else
            printf("Grade is 5");
    return 0;
}
```

(d)

```
#include<stdio.h>
int main()
{
    int s1,s2,s3,largest;
    scanf("%d %d %d",&s1,&s2,&s3);
    if(s1>s2 && s1>s3)
        largest=s1;
    else if(s2>s1 && s2>s3)
        largest=s2;
    else
        largest=s3;
    if((s1+s2)>largest || (s2+s3)>largest || (s1+s3)>largest)
        printf("Triangle is valid");
    else
        printf("Triangle is not valid");
    return 0;
}
```

(e)

```
#include<stdio.h>
int main()
{
    int x,y,z;
    scanf("%d %d %d",&x,&y,&z);
    if(x*x+y*y==z*z)
        printf("The triangle is right angle");
    else if((x==y) && (y==z))
    {
        printf("\nThe triangle is equilateral");
    }
    else if((x==z) && (x!=y) || (y==z) && (y!=x) || (x==y) && (x!=y))
    {
        printf("\nThe triangle is isoseles");
    }
    else
    {
        printf("\nThe triangle is scalene");
    }
    return 0;
}
```

(f)

```
#inlcude<stdio.h>
int main()
{
    int weigh;
    scanf("%d",&weigh);
    if(weigh<115)
        printf("Flyweight");
    else if(weigh>=115 && weigh<=121)
        printf("Bantamweight");
    else if(weigh>=122 && weigh<=153)
        printf("Featherweight");
    else if(weigh>=154 && weigh<=189)
        printf("Middleweight");
    else
        printf("Heavyweight");
    return 0;
}
```

(g)

```
#include<stdio.h>
int main()
{
    float r, g, b, c, m, y, k, w = 0;

    printf("\nEnter the color values of R, G and B : ");
    scanf("%f %f %f", &r, &g, &b);

    r /= 255;
    g /= 255;
    b /= 255;

    if(w < r)
        w = r;
```

```

if(w < g)
    w = g;

if(w < b)
    w = b;

c = (w - r) / w;
m = (w - g) / w;
y = (w - b) / w;
k = 1 - w;

printf("\nC : %f\nM : %f\nY : %f\nK : %f", c,m,y,k);
return 0;
}

```

(h)

```

#include<stdio.h>
int main()
{
    int d,m;

    printf("\nEnter the data and month of birth : ");
    scanf("%d %d", &d, &m);

    printf("\nYour Zodiac is : ");

    if(m == 12 && d >= 22 || m == 1 && d <= 19)
        printf("Capricorn.");

    if(m == 1 && d >= 20 || m == 2 && d <= 17)
        printf("Aquaries.");

    if(m == 2 && d >= 18 || m == 3 && d <= 19)
        printf("Pisces");

    if(m == 3 && d >= 20 || m == 4 && d <= 19)
        printf("Aries");

    if(m == 4 && d >= 20 || m == 5 && d <= 20)
        printf("Taurus");

    if(m == 5 && d >= 21 || m == 6 && d <= 20)
        printf("Gemini");

    if(m == 6 && d >= 21 || m == 7 && d <= 22)
        printf("cancer");

    if(m == 7 && d >= 23 || m == 8 && d <= 22)
        printf("Leo");

    if(m == 8 && d >= 23 || m == 9 && d <= 22)
        printf("Virgo");

    if(m == 9 && d >= 23 || m == 10 && d <=22)
        printf("Libra");

    if(m == 10 && d >= 23 || m == 11 && d <= 21)
        printf("Scorpio");

    if(m == 11 && d >= 22 || m == 12 && d <=21)
        printf("Sagittarius.");
    return 0;
}

```



```

    }
(i)
    #include<stdio.h>
    int main()
    {
        float w, h, bmi;

        printf("\nEnter you wight (in kg) and height (in m) : ");
        scanf("%f %f", &w, &h);

        bmi = w / (h * h);

        printf("\nYour BMI category is : ");

        if(bmi < 15)
            printf("Starvation");

        if(bmi >= 15.1 && bmi <= 17.5)
            printf("Anorexic");

        if(bmi >= 17.6 && bmi <= 18.5)
            printf("Underweight");

        if(bmi >= 18.6 && bmi <= 24.9)
            printf("Ideal");

        if(bmi >= 25 && bmi <= 25.9)
            printf("Overweight");

        if(bmi >= 30 && bmi <= 30.9)
            printf("Obese.");

        if(bmi >= 40)
            printf("Morbidly Obese");
        return 0;
    }

```

[E]

(a)

(1)

```

    #include<stdio.h>
    int main()
    {
        char ch;

        printf("Enter a character : ");
        scanf("%c", &ch);

        ch > 98 && ch < 123 ? printf("Lower case alphabet."):printf("Not a lower c
ase alphabet.");
        return 0;
    }

```

(2)

```

    #include<stdio.h>
    int main()
    {
        char chr;

```

```

printf("Enter a character : ");
scanf("%c", &chr);

ch < 123 && ch > 97 || chr > 64 && chr < 92 ? printf("Not a special symbol
.").):printf("Special symbol.");
return 0;
}

```

(b)

```

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

printf("Enter a year : ");
scanf("%d", &year);

year % 4 ? printf("%d is a not a leap year.", year):printf("%
d is a leap year.", year);
return 0;
}

```

OR

```

#include <stdio.h>
int main()
{
int year;
printf("Enter a year: ");
scanf("%d",&year);
if(year%4 == 0)
{
if( year%100 == 0)
{
// year is divisible by 400, hence the year is a leap year
if ( year%400 == 0)
printf("%d is a leap year.", year);
else
printf("%d is not a leap year.", year);
}
else
printf("%d is a leap year.", year );
}
else
printf("%d is not a leap year.", year);
return 0;
}

```

(c)

```

#include<stdio.h>
int main()
{
int a,b,c;

printf("Enter three numbers : ");
scanf("%d %d %d", &a,&b,&c);

a>b&&a>c?printf("%d is greatest", a):(b>a&&b>c?printf("%
d is greatest", b):printf("%d is greatest.",c));
return 0;
}

```

(d)

```

#include<stdio.h>
#include<math.h>
int main()
{
float theta;
printf("Enter angle in degrees\n");
scanf("%f",&theta);
theta=0.0174533*theta;
if((sin(theta)*sin(theta)+cos(theta)*cos(theta))==1)
{
    printf("\sin^2+cos^2=1\ is true");
}
else
{
    printf("\sin^2+cos^2=1\ is not true");
}
return 0;
}

```

(e)

```

#include<stdio.h>
int main()
{
float sal;
printf("Enter the salary");
scanf("%d",&sal);
sal>=25000 && sal<=40000?printf("Manager\n"):(sal>=15000 && sal<25000?
printf("Accountant\n");:printf("Clerk"));
return 0;
}

```

<https://github.com/hiverkiya/Let-Us-C-Solutions>

Loop Control Instruction

14 July 2020 16:10

Chapter-5 Loop Control Instruction

[A]

- (a) 1
2
3
4
5
6
7
8
9
10
- (b) 2 3 3
- (c) 3 3 1
- (d) prints it infinite times.
- (e) prints 10 infinite times.
- (f) prints nothing.

[B]

(a)

```
#include<stdio.h>
int main()
{
    int i, hour, extra;
    for(i = 1 ; i <= 10 ; i++)
    {
        printf("\n");
        printf("Enter the working hour of the employ : ");
        scanf("%d", &hour);

        if(hour>40)
        {
            extra = (hour - 40)*12;
            printf("\n%d Rs. is the overtime pay of employee\n", extra);
        }
        else
            printf("\nThis employ has not done overtime.\n");
    }
    return 0;
}
```

(b)

```
#include<stdio.h>
int main()
{
    int i, fact=1, num;
    scanf("%d", &num);
    printf("Enter a number = ");
    for(i=1; i<=num; i++)
    {
        fact*=num;
    }
}
```

```

    }
    printf("%d",fact);
    return 0;
}

```

(c)

```

#include<stdio.h>
#include<math.h>
int main()
{
    int base,power,d;
    print("Enter value of base = ");
    scanf("%d",&base);
    printf("Enter value of power = ");
    scanf("%d",&power);
    d=pow(base,power);
    printf("%d",d);
    return 0;
}

```

(d)

```

#include<stdio.h>
int main()
{
    int i;
    for(i=0;i<=255;i++)
        printf("\n%d = %c",i,i);
    return 0;
}

```

(e)

```

#include<stdio.h>
int main()
{
    int i,rem,num=0,x;
    for(i=1;i<=500;i++)
    {
        num=0;
        x=i;
        while(x)
        {
            rem=x%10;
            num=num+(rem*rem*rem);
            x/=10;
        }
        if(num==i)
            printf("%d is a armstrong number\n",i);
    }
    return 0;
}

```

(f)

```

#include<stdio.h>
main()
{
    int matchsticks=21, user, computer;
    printf("Do not enter Invalid Numbers.\nNumbers above 4 are invalid.");
    printf("\nIf you do so, the computer automatically wins.");
}

```

```

        while (matchsticks>=1)
        {
            printf("\nNumber of matchsticks available right now is %
d.", matchsticks);
            printf("\n\nYour Turn...\n\n\n");
            printf("\nPick up the matchstick(s)-- (1-4): ");
            scanf ("%d", &user);
            if (user>4)
            {
                printf("Invalid Selection");
                break;
            }
            computer=5-user;
            printf("\nComputer's Turn..\n" );
            printf("\nComputer chooses:%d", computer);

            matchsticks=matchsticks-user-computer;
            continue;
            if(matchsticks==1)
            break;
        }
        matchsticks--;
        printf("\nComputer Wins");
    }
}

```

OR

```

#include
#include
void main()
{
    int matchsticks=21, user, computer;
    clrscr();
    printf("Do not enter Invalid Numbers.\nNumbers above 4 are invalid.");
    printf("\nIf you do so, the computer automatically wins.");
    while (matchsticks>=1)
    {
        printf("\nNumber of matchsticks available right now is %
d.", matchsticks);
        printf("\n\nYour Turn...\n\n\n");
        printf("\nPick up the matchstick(s)-- (1-4): ");
        scanf ("%d", &user);
        if (user>4)
        {
            printf("Invalid Selection");
            break;
        }
        computer=5-user;
        printf("\nComputer's Turn..\n" );

        if(matchsticks==1)
        break;
        printf("\nComputer chooses:%d", computer);
        matchsticks=matchsticks-user-computer;
        continue;
    }
    matchsticks--;
    printf("\nComputer Wins");
    getch();
}
(g)

```

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

```

int main()
{
int num,pos=0,neg=0,zero=0,choice;
do
{
scanf("%d",&num);
if(num==0)
zero++;
else if(n<0)
neg++;
else if(n>0)
pos++;
printf("Do you want to enter another number? (Enter 1 for yes & 0 f
or no)");
scanf("%d",&choice);
}while(choice);
printf("Zeroes = %d\n",zero);
printf("Negatives = %d\n",neg);
printf("Positives = %d\n",pos);
return 0;
}
(h)

```

```

#include<stdio.h>
int main()
{
int num,rem,oct=0,rev=0,rem1;
scanf("%d",&num);
while(num)
{
rem=num%8;
oct=10*oct+rem;
num/=8;
}
while(oct)
{
rem1=oct%10;
rev=rev*10+rem1;
oct/=10;
}
printf("%d is octal equivalent",rev);
return 0;
}

```

OR

```

#include<stdio.h>
int main()
{
int number;
printf("Enter Number\n");
scanf("%d",&number);
printf("Octal equivalent is %o",number);
return 0;
}

```

(i)

```

#include<stdio.h>
int main()
{
int i,num,max,min, N, range;

```

```
printf("Enter how many number you want to enter : ");
scanf("%d", &N);

printf("Enter the number : ");
scanf("%d", &num);

max=min=num;

for( i = 1; i < N; i++)
{
    printf("Enter the number : ");
    scanf("%d", &num);
    if(num>max)
        max=num;
    if(num<min)
        min=num;
}
range = max - min;

printf("\n%d is the range of the data.", range);
return 0;
}
```

<https://github.com/hiverkiya/Let-Us-C-Solutions>

More Complex Repetitions

14 July 2020 16:10

Chapter-6 More Complex Repetitions

[A]

- (a) prints nothing.
- (b) 1 (infinite times).
- (c) 2
5

[B]

- (a) the initialization expression.
the testing expression.
the increasing/decreasing expression.
- (b) a for loop.
- (c) At least once.
- (d) Initialization, execution of body, testing.
- (e) 3.
- (f) Continue;
- (g) Infinite times.
- (h) The program will produce the output $x = 10$ $y = 10$.
The ; after the if ($x \neq y$) would NOT produce an error.
- (i) All things that can be done using a for loop can also be done using a while loop.
for(;;) implements an infinite loop.

[C]

(a)

```
#include<stdio.h>
int main()
{
    int i,j,flag;
    for(i=1;i<=300;i++)
        flag=1;
    for(j=2;j<=i/2;j++)
    {
        if(i%j==0)
        {
            flag=0;
            break;
        }
    }
    if(flag==1)
        printf("%d\n",i);
}
return 0;
}
```

(b)

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

```

{
    int i=1;
    while(i)
    {
        printf("%c",i);
    }
    return 0;
}

```

(c)

```

#include<stdio.h>
int main()
{
    int i,j,fact;
    float sum=0,d;
    for(i=1;i<=7;i++)
    {
        fact=1;
        for(j=1;j<=i;j++)
        {
            fact=fact*j;
        }
        d=i/fact;
        sum=sum+d;
    }
    printf("Sum of first seven terms is = %d",sum);
    return 0;
}

```

(d)

```

#include<stdio.h>
int main()
{
    int i,j,k;
    for(i=1;i<=3;i++)
    {
        for(j=1;j<=3;j++)
        {
            for(k=1;k<=3;k++)
            {
                if(i==j || i==k || j==k)
                    continue;
                printf("%d, ", i*100+j*10+k);
            }
        }
    }
    printf("are the all possible combinations of 1,2 and 3.");
    return 0;
}

```

(e)

```

#include<stdio.h>
#include<stdlib.h>
int main()
{
    int year=0,inv,alt;
    while(alt>inv)
    {
        year++;
        alt=120*year;
        inv=(1000*year)-4000;
    }
}

```

```

printf("The minimum year is %d",year);
}
(f)
#include<stdio.h>
int main()
{
    int n,i,mul;
    printf("Enter a number = ");
    scanf("%d",&n);
    for(i=1;i<=10;i++)
    {
        mul=n*i;
        printf("%d * %d = %d\n",n,i,mul);
    }
    return 0;
}

(g)
#include<stdio.h>
int main()
{
    int y;
    float x,i;
    printf("                y                x                =                i");

    for(y=1;y<=6;y++)
    {
        for(x=5.5;x<=12.5;x+=0.5)
        {
            i=2+(y+(x*0.5));
            printf("\n                %d                %f                =                %f", y, x, i);

        }
    }
    return 0;
}

(h)
#include<stdio.h>
#include<math.h>
int main()
{
    float i,p,q,r,n,a,d;
    for(i=1;i<=10;i++)
    {
        printf("\n
\nEnter principal, rate, time (in year) and compound interest respectively : "
);
        scanf("%f %f %f %f", &p,&r,&n,&q);

        d=pow((1+r/q),n*q);
        a=p*d;

        printf("\n%f is the amount.", d);
    }
    return 0;
}

(i)
#include<stdio.h>
#include<math.h>
int main()

```

```

{
    int a,x,i,j,term;
    float sol,sum;
    printf("Enter the value of x = ");
    scanf("%d",&x);
    term=(x-1)/x;
    sum=term;
    int power=2;
    for(i=1;i<=6;i++)
    {
        sol=pow(term,a)/2;
        sum=sum+sol;
        a++;
    }
    printf("%f is the sum of the series.\n", sum);
    return 0;
}

```

(j)

```

#include<stdio.h>
int main()
{
    int i, j, k;
    for(i = 1; i <= 30; i++)
    {
        for (j = i; j <= 30; j++)
        {
            for (k = j; k <= 30; k++)
            {
                if (i*i + j*j == k*k || j*j + k*k == i*i || i*i + k*k == j*j)
                    printf("\n%d %d and %d", i, j, k);
            }
        }
    }
    printf(" are all pythagorian triplets.\n");
    return 0;
}

```

(k)

```

#include<stdio.h>
int main()
{
    int population = 100000;
    for(int i = 0; i < 10; i++)
    {
        population = population - (population / 100) * 10;
        printf("Year %d : %d\n", 10-i, population );
    }
    return 0;
}

```

(l)

```

#include<stdio.h>
#include<conio.h>
#include<math.h>
int main()
{
    int num, i, j, k, l;
    for (num = 1; num < 5000; num++)
    {
        for (i = 1; i < num; i++)
        {

```

```

        if (num < i*i*i)
            break;
    or (j = i + 1; j < num; j++)
    {
        if (num < j*j*j)
            break;
    for (k = i + 1; k < num; k++)
        {
            if (k*k*k > i*i*i + j*j*j)
                break;
            for (l = k + 1; l < num; l++)
                {
                    if (num < k*k*k + l*l*l)
                        break;
                    if ((num == i*i*i + j*j*j) && (num == k*k*k + l*l*l))
                {
                    printf("\n%d^3 + %d^3 = %d^3 + %d^3 = num : %d",i, j, k, l, num);
                    break;
                }
            }
        }
    }
}
getch();
return 0;
}
(m)
#include<stdio.h>
#include<conio.h>
int main()
{
    int hr;
    for(hr = 0; hr < 24; hr++)
    {
        if(hr == 0)
            printf("\n12 Midnight");
        if(hr > 0 && hr < 12)
            printf("\n%d AM", hr);
        if(hr == 12)
            printf("\n%d Noon.", hr);
        if(hr > 12 && hr < 24)
            printf("\n%d PM.", hr-12);
    }
    getch();
    return 0;
}
(n) Floyd's Triangle
#include<stdio.h>
int main()
{
    int num,r,c,sp,i=1;
    printf("Enter any number : ");
    scanf("%d", &num);
    for(r=1; r<=num; r++)
    {
        for(sp=1; sp<=num-r; sp++)
            printf(" ");
        for(c=1; c<=r; c++,i++)
            printf("%d ",i);
        printf("\n");
    }
}

```

```

        return 0;
    }
(o)

void main()
{
    int i,j,k;
    for(i=7;i>=1;i--)
    {
        for(j=1;j<=7;j++)
        {
            if(j<=i)
                printf("%c", 'A' + j-1);
            else
                printf(" ");
        }
        for(j=6;j>=1;j--)
        {
            if(j<=i)
                printf("%c", 'A' + j-1);
            else
                printf(" ");
        }
        printf("\n");
    }
}

```

(p)

Pascal's triangle is a triangular array of the binomial coefficients.

```

#include <stdio.h>
int main()
{
    int rows, coef = 1, space, i, j;
    printf("Enter number of rows: ");
    scanf("%d",&rows);
    for(i=0; i<rows; i++)
    {
        for(space=1; space <= rows-i; space++)
            printf(" ");
        for(j=0; j <= i; j++)
        {
            if (j==0 || i==0)
                coef = 1;
            else
                coef = coef*(i-j+1)/j;
            printf("%4d", coef);
        }
        printf("\n");
    }
    return 0;
}

```

<https://github.com/hiverkiya/Let-Us-C-Solutions>

Case Control Instructions

14 July 2020 16:11

[Chapter-7 Case Control Instructions] Solutions

[A]

- (a) Heart
I thought one wears a suite.
- (b) You always lose gold prize.
- (c) At least employees are good.
- (d) Trapped.
- (e) You entered a and b.
- (f) Feeding fish
Weeding grass
Mending roof
Just to survive

[B]

(a) Expected ':' instead of ';' in case statements because with ';' case statements do not belong to any switch.

(b) temp can't appear in constant expression. We can never have a variable in case statement, but temp is there.

(c) Here 'a' is a float variable and we cannot test floats in switch statements.

(d) 'a' and 'b' are not constants used in case statement.

[C]

```
#include<stdio.h>
int main()
{
    int choice,num,i,j,fact=1,isPrime=0;
    printf("Enter your choice = ");
    printf("press 1 for factorial of a number");
    printf("press 2 to check prime or not");
    printf("press 3 to check odd or even");
    scanf("%d",&choice);
    switch(choice)
    {
        case 1:
            printf("Enter a number = ");
            scanf("%d",&num);
            for(i=1;i<=num;i++)
                fact*=num;
            printf("Factorial of a number is %d",fact);
            break;
        case 2:
            printf("Enter a number = ");
            scanf("%d",&num);
            for(i=2; i<=num/2; ++i)
            {
                if(num%i==0)
                {
```

```

        isPrime=1;
        break;
    }
}
if (isPrime==0)
    printf("%d is a prime number.",num);
else
    printf("%d is not a prime number.",num);
    break;
case:3
    printf("Enter a number = ");
    scanf("%d",&num);
    if(num%2==0)
        printf("%d is odd",num);
    else
        printf("%d is even",num);
        break;
case:4
    printf("Thanks for using");
    exit(1);
    break;
}
if(choice!=1 || choice!=2 || choice!=3 || choice!=4)
    printf("You entered a wrong choice. Try Again!!");
    return 0;
}

```

OR

```

#include<stdio.h>
#include<conio.h>
int main()
{
int a,x,num,p,fact,prm;
for(a=2;a>1;a++)
{
    printf("\n\nEnter what your choice.\n");
    printf("\n\n1. Find factorial.");
    printf("\n\n2. Check Prime number.");
    printf("\n\n3. Check even and odd.\n\n4. Exit.\n");
    scanf("%d", &x);
    switch(x)
    {
    case(1):
        printf("Enter the number of what you want to find the factorial :");
        scanf("%d", &num);
        fact=num;
        for(p=1;p<num;p++)
            fact=fact*p;
        printf("%d is the factorial.", fact);
        break;

    case(2):
        printf("Enter the number which you want to check whether it is prime or not : ");
        scanf("%d", &num);
        for(;p<num;p++)
        {
            prm=num%p;
            if(prm==0)
            {

```



```

        printf("%d is not a prime number.", num);
        break;
    }
    if(p==num-1)
        printf("%d is a prime number.", num);
    }
    break;

case(3):
    printf("Enter the number to check whether it is even or odd : ");
    scanf("%d", &num);
    if(num%2==0)
        printf("%d is an even number.", num);
    else
        printf("%d is odd odd number.", num);
    break;
}

if(x==4)
{
    printf("\n\nOk Thank you for using this program.\n\n Bye.\n\n\n Exiting.");
    break;
}
if(x!=1 && x!=2 && x!=3 && x!=4)
    printf("You have entered the wrong choice. Try Again!!");
}
getch();
return 0;
}

```

[D]

```

#include<stdio.h>
int main()
{
    int class,fail;
    printf("Enter the class = ");
    scanf("%d",&class);
    printf("Enter the number of subjects student got failed = ");
    scanf("%d",&fail);
    switch(class)
    {
        case 1:
            switch(fail)
            {
                case 0:
                case 1:
                case 2:
                case 3:
                    printf("You've got grace of 5 marks per subject");
                    break;
                default:
                    printf("You didn't got any grace");
                    break;
            }
            break;
        case 2:
            switch(fail)
            {
                case 0:
                case 1:
                case 2:
                    printf("You've got grace of 4 marks per subject");

```

```

        break;
        default:
        printf("You didn't got any grace");
        break;
    }
    break;
case 3:
    switch(fail)
    {
        case 0:
        case 1:
        printf("You've got grace of 3 marks per subject");
        break;
        default:
        printf("You didn't got any grace");
        break;
    }
    break;
    default:
    printf("Wrong Choice. Please try again!!");
    break;
}
return 0;
}

```

<https://github.com/hiverkiya/Let-Us-C-Solutions>

Functions

14 July 2020 16:11

Chapter-8 Functions

[A]

- (a) Learn C
Followed by C++, C# and Java
(infinite times)
- (b) 100
- (c) 3.140000
- (d) Error.

[B]

- (a) Semicolor missing in the declaration of a function.
A function cannot return more than one values.
- (b) message function has return type void, so it cannot be assigned to a ny variable.
- (c) Function definition argument should have a datatype.
Function is not defined before calling, or there should be prototyp e declaration of the function.
- (d) Invalid use of a semicolon.
- (e) Function definition is invalid in other functions.
- (f) void is send in the message as argument, which is invalid.

[C]

- (a) No! There is an Invalid use of a semicolon.
- (b)
 - (1) False.
 - (2) False.
 - (3) True.
 - (4) False.
 - (5) True.
 - (6) True.
 - (7) True.
 - (8) False.
 - (9) True.
 - (10) False.

[D]

- (a)

```
#include<stdio.h>
int fact(int );
int main()
{
    int num,fac;
    scanf("%d",&num);
    fac=fact(num);
    printf("%d",fac);
    return 0;
```

```

    }
    int fact(int n)
    {
        int i,factorial=1;
        for(i=1;i<=n;i++)
        {
            factorial*=i;
        }
        return factorial;
    }

```

(b)

```

#include<stdio.h>
#include<math.h>
int power(int,int);
int main()
{
    int a,b,ans;
    scanf("%d %d",&a,&b);
    ans=power(a,b);
    printf("%d",ans);
    return 0;
}
int power(int a,int b)
{
    int d;
    d=pow(a,b);
    return d;
}

```

OR

```

#include<stdio.h>
#include<math.h>
int power(int,int);
int main()
{
    int a,b,ans;
    scanf("%d %d",&a,&b);
    ans=power(a,b);
    printf("%d",ans);
    return 0;
}

```

```

int power(int x, int y)
{
    int num = 1, i;
    for (i = 1; i <= y; i++)
        num = num*x;
    return(num);
}

```

(c)

```

#include<stdio.h>
#include<conio.h>
void rom(int);
void print(int, char);
int main()
{
    int yer;
    printf("Enter the year : ");
    scanf("%d", &yer);
}

```

```

printf("\nRoman equivalent of %d is : ", yer);
rom(yer);
getch();
return 0;
}
void rom(int yer)
{
int v, x, l, c, d, m;
m = yer / 1000;
print(m, 'm');
yer %= 1000;
d = yer / 500;
print(d, 'd');
yer %= 500;
c = yer / 100;
print(c, 'c');
yer %= 100;
l = yer / 50;
print(l, 'l');
yer %= 50;
x = yer / 10;
print(x, 'x');
yer %= 10;
v = yer / 5;
print(v, 'v');
yer %= 5;
print(yer, 'i');
}
void print(int a, char c)
{
int i;
for (i = 0; i < a; i++)
printf("%c", c);
}

```

(d)

```

#include<stdio.h>
int calc(int );
int main()
{
int year, leap;
printf("Enter year = ");
scanf("%d", &year);
leap = calc(year);
if (leap == 1)
printf("%d is leap year", year);
else
printf("%d is not a leap year", year);
return 0;
}
int calc(int year)
{
if (year % 4 == 0 || year % 100 == 0 || year % 400 == 0)
return 1;
else
return 0;
}

```

(e)

```

#include<stdio.h>
void prime(int );
int main()
{
int num;

```

```

        printf("Enter a number = ");
        scanf("%d",&num);
        prime(num);
        return(0);
    }
    void prime(int n)
    {
        int i,j,isPrime=1;
        for(i=2;i<=n;i++)
        {
            if(n%i==0)
            {
                isPrime=1;
                for(j=2;j<=i/2;j++)
                {
                    if(i%j==0)
                    {
                        isPrime=0;
                        break;
                    }
                }
                if(isPrime==1)
                    printf("%d\n",i);
            }
        }
        return 0;
    }
}

```

OR

```

#include<stdio.h>
#include<conio.h>
int check_prime(int);
void pf(int);
int main()
{
    int num;
    printf("Enter a number : ");
    scanf("%d", &num);
    pf(num);
    _getch();
    return 0;
}
void pf(int num)
{
    int i;
    for (i = 2; num != 1; i++) //loop for obtaining the numbers.
    {
        if (num%i != 0)// If the number is not a factor
            continue;
        else
        {
            //If the number is prime, so check if it is prime
            if(check_prime(i) == 1)
            {
                while (num%i == 0)
                {
                    printf("%d, ", i);
                    num /= i;
                }
            }
        }
    }
}

```

```
}  
//returns 1 if prime, otherwise 0.  
int check_prime(int num)  
{  
    int i = 2;  
    while(i < num)  
    {  
        if(i%num == 0)  
            return 0;  
        i++;  
    }  
    if(i == num)  
        return 1;  
}
```

<https://github.com/hiverkiya/Let-Us-C-Solutions>

Pointers

14 July 2020 16:11

[Chapter 9: Pointers] Solutions

[A]

(a) 5 2

(b) 25 4

(c)

1006 1006 1006

13.5 13.5 13.5 13.5 13.5

[B]

(a) Error:

1. pass() cannot return a value as its return type is void.
2. we cannot assign pass() to other variable as its return type is void
3. We cannot define variables inbetween function name and its body.

(b) Error: Invalid conversion fro int to int*. Cannot pass address in the function whose argument accepts integer.

(c) Error:

1. Can't assign a function returning void, to other variable.
2. m is not declared in the argument declaration feild of function.

(d) Error:

1. Can't assign a function returning void, to other variable.
2. function() cannot return anything with return type void.

[C]

(a)

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
void calc(int, int, int, int, int, int*, int*, float*);
int main()
{
    int a, b, c, d, e, sm, av;
    float sd;
    printf("Enter five numbers : ");
    scanf("%d%d%d%d%d", &a, &b, &c, &d, &e);
    calc(a, b, c, d, e, &sm, &av, &sd);
    printf("\nSum : %d\n\nAverage : %d\n\nStandard Deviation : %f\n", sm, av, sd);
    getch();
    return 0;
}
void calc(int a, int b, int c, int d, int e, int *sm, int *av, float *sd)
{
    int i;
    *sm = a + b + c + d + e;
    *av = *sm / 5;
    *sd = pow(a - *av, 2) + pow(b - *av, 2) + pow(c - *av, 2) + pow(d - *av, 2)
) + pow(e - *av, 2);
    *sd = sqrt(*sd / 5);
}
```



```
}
```

(b)

```
#include<stdio.h>
#include<conio.h>
void calc(int, int, int, float*, float*);
int main()
{
    float av, pr;
    int a, b, c;
    printf("Enter the marks of the student in three subjects out of 100 :");
    scanf("%d%d%d", &a, &b, &c);
    calc(a, b, c, &av, &pr);
    printf("\n\nAverage of the marks : %f\n\nPercentage of the student : %f", av, pr);
    getch();
    return 0;
}
void calc(int a, int b, int c, float *av, float *pr)
{
    *av = (a + b + c) / 3;
    *pr = *av / 3;
}
```

(c)

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
int fact(int);
float sine(float value_at);
int main()
{
    float x, sum;
    printf("Enter the angle in radians : ");
    scanf("%lf", &x);
    sum = sine(x); //calling the sum of he series
    printf("\n\nsin(%lf) = %lf", x, sum);
    getch();
    return 0;
}
float sine(float x)
{
    int i, j, sign = 1;
    float sum = 0;
    for (i = 0, j = 1; i < 5; i++, j += 2, sign *= -1)
        sum = sum + /*pow(-1, i)*/sign*pow(x, j) / fact(j);
    return sum;
}
int fact(int num)
{
    int ans = 1;
    while (num > 0)
    {
        ans *= num;
        num--;
    }
    return ans;
}
```

(d)

```

#include<stdio.h>
#include<conio.h>
void chng(int*, int*, int*);
int main()
{
    int a, b, c;
    printf("Enter three numbers : ");
    scanf("%d%d%d", &a, &b, &c);
    printf("\n\nYou've Entered\n\n a : %d    b : %d    c : %d\n", a, b, c);
    chng(&a, &b, &c);
    printf("\n\nAfter Shifting\n\n a : %d    b : %d    c : %d\n", a, b, c);
    getch();
    return 0;
}
void chng(int *a, int *b, int *c)
{
    int x;
    x = *c;
    *c = *b;
    *b = *a;
    *a = x;
}

```

(e)

```

#include<stdio.h>
#include<conio.h>
#include<math.h>
float ar(float, float, float);
int main()
{
    float area, a, b, c;
    printf("Enter the sides of the triangle : ");
    scanf("%f%f%f", &a, &b, &c);
    if (a + b > c && b + c > a && a + c > b)
    {
        area = ar(a, b, c);
        printf("\n\nArea of the triangle : %f", area);
    }
    else
        printf("\n\nThe triangle you entered is invalid.");
    getch();
    return 0;
}
float ar(float a, float b, float c)
{
    float area;
    float S = (a + b + c) / 2;
    area = sqrt(S*(S - a)*(S - b)*(S - c));
    return (area);
}

```

(f)

```

#include<stdio.h>
#include<conio.h>
#include<math.h>
float dis(int, int, int, int);
float ar(int, int, int, int, int, int);
int main()
{

```

```

int x1, x2, x3, y1, y2, y3, x, y;
float A1, A2, A3, A;
printf("Enter the coordinates of first point of the triangle : ");
scanf("%d%d", &x1, &y1);
printf("\n\nEnter the coordinates of second point of the triangle : ");
scanf("%d%d", &x2, &y2);
printf("\n\nEnter the coordinates of third point of the triangle : ");
scanf("%d%d", &x3, &y3);
printf("\n\nEnter the points to check its position : ");
scanf("%d%d", &x, &y);
A1 = ar(x1, y1, x2, y2, x, y);
printf("A1 : %f\n", A1);
A2 = ar(x1, y1, x3, y3, x, y);
printf("A2 : %f\n", A2);
A3 = ar(x, y, x2, y2, x3, y3);
printf("A3 : %f\n", A3);
A = ar(x1, y1, x2, y2, x3, y3);
printf("A : %f\n", A);
if (!(A1 + A2 + A3 > A || A1 + A2 + A3 < A))
    printf("\n\nThe point (%d,%d) lies inside of triangle.", x, y);
else
    printf("\n\nThe point (%d,%d) lies outside of triangle.", x, y);
getch();
return 0;
}
float dis(int x1, int y1, int x2, int y2)
{
    float distance;
    distance = sqrt(pow((x2 - x1), 2) + pow((y2 - y1), 2));
    return (distance);
}
float ar(int x1, int y1, int x2, int y2, int x3, int y3)
{
    float a, b, c, area, S;
    a = dis(x1, y1, x2, y2);
    b = dis(x1, y1, x3, y3);
    c = dis(x2, y2, x3, y3);
    S = (a + b + c) / 2;
    area = sqrt(S*(S - a)*(S - b)*(S - c));
    return (area);
}

```

(g)

```

#include<stdio.h>
#include<conio.h>
int gcd(int, int);
int main()
{
    int a, b, cd, max, min;
    printf("Enter two numbers : ");
    scanf("%d%d", &a, &b);
    if (a>b)//for making a greater number
    {
        max = a;
        min = b;
    }
    else
    {
        max = b;
        min = a;
    }
}

```

```
a = max;
b = min;
cd = gcd(a, b); //returning the greatest divisor
printf("\n\nGreatest common divisor of the givn numbers is %d", cd);
getch();
return 0;
}
int gcd(int a, int b)
{
    static int x, temp;
    if (b == 0)
        return (a);
    x = a / b;
    temp = a;
    a = b;
    b = temp - x*b;
    gcd(a, b);
}
```

<https://github.com/hiverkiya/Let-Us-C-Solutions>

Recursion

14 July 2020 16:11

[Chapter 10: Recursion] Solutions

[A]

- (a) Infinite Print C adds wings to your thoughts.
- (b) C adds wings to your thoughts.

[B]

(a)

```
#include<stdio.h>
#include<conio.h>
int nonrec_calc(int);
int rec_calc(int);
int main()
{
    int num, sum;
    printf("Enter a five digit number : ");
    scanf("%d", &num);
    sum = nonrec_calc(num);
    printf("\n\nSum of digits using non-recursive function : %d\n", sum);

    sum = rec_calc(num);
    printf("\n\nSum of digits using recursive function: %d\n", sum);

    _getch();
    return 0;
}
//Non recursive function
int nonrec_calc(int num)
{
    int a, sum = 0, i;
    for (i = 0; num; i++)
    {
        a = num % 10; //obtaining the ladt digit
        sum = sum + a; //summing up the digits
        num = num / 10; //decreasing the number by one digit
    }
    return (sum);
}
//Recursive function
int rec_calc(int num)
{
    int sum = 0;
    if (num == 0)
        return sum;
    sum = num % 10 + rec_calc(num / 10);
    return sum;
}
```

(b)

```
int func(int num);
#include <stdio.h>
int main()
{
    int number;
    printf("Enter number\n");
```

```

scanf("%d",&number);
func(number);
return 0;
}
int func(int num)
{
int i;
for(i=2;i<=num;i++)
{
    if(num%i==0)
    {
        printf("%d\n",i);
        func(num/i);
        break;
    }
}
return 0;
}

```

(c)

```

#include<stdio.h>
#include<conio.h>
void fs(int first, int second, int term);
int main()
{
fs(0, 1, 25);
_getch();
return 0;
}
void fs(int fis, int sec, int term)
{
int num;
if (term == 0)
return;
num = fis + sec;
fis = sec;
sec = num;
printf("%d, ", num);
fs(fis, sec, term - 1);
}

```

(d)

(1) Without Recursion

```

void nonrec_bin(int);
#include<stdio.h>
int main()
{
    int number;
    printf("Enter number\n");
    scanf("%d",&number);
    nonrec_bin(number);
    return 0;
}
void nonrec_bin(int num)
{
int sum = 0;
for (; num; num /= 2)
{
    //Next digit will placed at first position.

```

```

    sum += num % 2;
    sum *= 10;
}
printf("%d", sum);
}

```

(2) With recursion

```

#include<stdio.h>
int bin(int);
int main()
{   int number;
printf("Enter number\n");
scanf("%d",&number);
bin(number);
return 0;
}
int bin(int number)
{
if(number>1)
{
    bin(number/2);
}
printf("%d",number%2);
return 0;
}

```

(e)

```

int sum(int);
int main()
{   int number,ans;
printf("Enter number\n");
scanf("%d",&number);
ans=sum(number);
printf("%d",ans);
return 0;
}

int sum(int x)
{
if (x == 0)
return x;
x = x + sum(x - 1);
return x;
}

```

<https://github.com/hiverkiya/Let-Us-C-Solutions>

Data Types Revisited

14 July 2020 16:12

[Chapter 11: Data Type Revisited] Solutions

[A]

(a) Will Print Numbers from 1 to 50000, separated by newline.

(b) 13.500000 13.500000

(c) main's i=0 val's i=100 main's i= 101 val's i=100

(d) 6 5 6

(e) count=5
count=4
count=3
count=2
count=1

(f) 6
9
13
18

(g) 1 1 1
1 1 2

(h) 30
20

[B]

(a) NO ERROR.

(b) NO ERROR. (Warning: As saving integer to a char type variable, it'll truncate value and store)

(c) NO ERROR.

(d) Double cannot be unsigned, and float "out of range" warning.

(e) NO ERROR.

[C]

(a) True

(b) False

(c) False

(d) False

(e) False

(f) False

(g) False

(h) False

(i) False

(j) False

(k) True

(l) True

(m) False

<https://github.com/hiverkiya/Let-Us-C-Solutions>

C preprocessor

14 July 2020 16:12

[Chapter 12: The C Preprocessor] Solutions

[A]

- (a) 3. a message from programmer to the preprocessor
- (b) `#define SQR(X) (X * X)`
- (c)
 - 1. False
 - 2. False
 - 3. True
 - 4. False
 - 5. False
 - 6. False
- (d) We can include any number of files we want to include in a program.
- (e) In the first statement, we can give any path of the file to be included. While the name given in the second statement looks for the file in the default set of directories.
- (f) All are true.
- (g) `#elseif`
- (h) 1. Before the compilation of program.
- (i) 1. Before the compilation of program.

[B]

- (a) 2
- (b) 9 49
- (c)
 - `a = 43.14`
 - `b = 5863.727051`

[C]

- (a)

Before compilation, our source code is expanded and stored in the file `FILENAME.I`, so we can open this file and check how our program is getting expand.
- (b)
 - 1. `#define LOWER (alpha>=97 && alpha<=122)`
 - 2. `#define UPPER (alpha>=65 && alpha<=90)`
 - 3. `#define UPPER (alpha>=65 && alpha<=90)`
`#define LOWER (alpha>=97 && alpha<=122)`
`#define YES (UPPER || LOWER)`
 - 4. `#define MAX(a,b) ((a>b)?a:b)`
- (c) Code for header file `areaperi.h`

`#include<math.h>`
`#define PI 3.14`

```

#define S(a,b,c) ((a+b+c)/2.0)
#define TRI_AREA(a,b,c) (sqrt((S(a,b,c))*((S(a,b,c))-a)*((S(a,b,c))-
b)*((S(a,b,c))-c)))
#define TRI_PERI(a,b,c) (a+b+c)
#define SQR_AREA(x) (x*x)
#define SQR_PERI(x) (2*(x+x))
#define CIR_AREA(r) (PI*r*r)
#define CIR_PERI(r) (2*PI*r)

```

Actual Program

```

#include<stdio.h>
#include<conio.h>
#include "areaperi.h"
int main()
{
    int r, a, x, y, z;
    double cir_ar, cir_peri, sqr_ar, sqr_peri, tri_ar, tri_peri;
    printf("Enter the radius of the circle : ");
    scanf("%d", &r);
    printf("\nEnter the side of the square : ");
    scanf("%d", &a);
    printf("\nEnter the sides of the triangle : ");
    scanf("%d%d%d", &x, &y, &z);
    if (x + y > z && y + z > x && x + z > y)
    {
        tri_ar = TRI_AREA(x, y, z);
        tri_peri = TRI_PERI(x, y, z);
        printf("\nTriangle");
        printf("\nArea : %lf\nPerimeter : %f\n", tri_ar, tri_peri);
    }
    else
        printf("\nThe triangle You entered is invalid.\n");
    cir_ar = CIR_AREA(r);
    cir_peri = CIR_PERI(r);
    sqr_ar = SQR_AREA(a);
    sqr_peri = SQR_PERI(a);
    printf("\nCircle");
    printf("\nArea : %f\nPerimeter : %f\n", cir_ar, cir_peri);
    printf("\nSquare");
    printf("\nArea : %f\nPerimeter : %f\n", sqr_ar, sqr_peri);
    getch();
    return 0;
}
(d)

```

- (1) #define MEAN(x,y) ((x+y)/2.0)
- (2) #define ABS(x) ((x<0)?(-1*x):x)
- (3) #define U2L(c) (c+32)
- (4) #define MAX(a,b) ((a>b)?a:b)

(e) Header file code

```

#define SI(a,b,c) ((a*b*c)/100.0)
#define AMOUNT(si,p) (si+p)

```

Actual file code

```

#include<stdio.h>
#include<conio.h>
#include "interest.h"
int main()

```

```
{
float p, t, r, si, amnt;
printf("Enter the principal, time in year and rate of interest : ");
scanf("%f%f%f", &p, &t, &r);
si = SI(p, t, r);
amnt = AMOUNT(si, p);
printf("\nSimple Interest : %f\nAmount : %f", si, amnt);
_getch();
return 0;
}
```

<https://github.com/hiverkiya/Let-Us-C-Solutions>

Arrays

14 July 2020 16:13

Chapter-13 Arrays

[A]

- (a) 200 100
- (b) ASCII Value Table
- (c) 49

[B]

- (a) int followed by char is illegal
No '&' sign in scanf function
- (b) No & sign in scanf function
- (c) No error.

[C]

(a) An array is a collection of the same data type placed next to each other in memory.

- (b) No: Array index is comes inside [].
No: Array index must have a constant, here size is a variable.
No: c is an int, and it cannot contain an array

(c) Third Element.

(d) first is array size, second is particular element.

(e)

- (1) True.
- (2) False.
- (3) False.
- (4) True.

[D]

(a)

```
#include<stdio.h>
int main()
{
    int i,count=0,num;
    int a[25];
    for(i=0;i<25;i++)
    {
        scanf("%d",&a[i]);
    }
    scanf("%d",&num);
    for(i=0;i<25;i++)
    {
        if(num==a[i])
            count++;
    }
    if(count>0)
        printf("The number is present %d times",count);
    else
        printf("Number doesn't exit in array");
    return 0;
}
```

(b)

```
#include<stdio.h>
#define MAX 5
void selection(int*);
void bubble(int*);
void insertion(int*);
void swap(int*, int*);
void show(int*); //display the array.
int main()
{
    int a[MAX] = { 5,1,4,2,7 }, b[MAX] = { 5, 2, 3, 9, 0 }, c[MAX] = {6, 23,1,
6, 2};
    printf("\nSelection sort.");
    selection(a);
    show(a);
    printf("\nBubble sort.");
    bubble(b);
    show(b);
    printf("\nInsertion sort.");
    insertion(c);
    show(c);
    return 0;
}

void selection(int *a)
{
    int i, j;
    for (i = 0; i < MAX; i++)
        for (j = i + 1; j < MAX; j++)
            if (a[i] > a[j])
                swap(&a[i], &a[j]);
}

void bubble(int *a)
{
    int i, j;
    for (i = 0; i < MAX; i++)
        for (j = 0; j < MAX - 1; j++)
            if (a[j] > a[j + 1])
                swap(&a[j], &a[j + 1]);
}

void insertion(int *a)
{
    int key, j, k;
    /* Grabs the next key on each iteration. */
    for (int i = 1; i < MAX; i++)
    {
        /* Selecting the number. */
        key = arr[i];
        /* Finding the position to insert viz. j */
        for (j = 0; arr[j] < key; j++);
        /* Shifting numbers to left. */
        for (k = i; k > j; k--)
            arr[k] = arr[k - 1];
        /* Inserting the number. */
        arr[j] = key;
    }
}

void swap(int *a, int *b)
{
    int temp;
    temp = *a;
    *a = *b;
    *b = temp;
}
```

```

    }
    void show(int *a)
    {
        int i;
        printf("\n");
        for (i = 0; i < MAX; i++)
            printf("%d\t", a[i]);
        printf("\n");
    }

```

(c)

```

#include<stdio.h>
int main()
{
    int i,j,isPrime;
    int a[100];
    for(i=0;i<100;i++)
        a[i]=i+1;
    for(i=0;i<100;i++)
    {
        isPrime=1;
        for(j=2;j<a[i];j++)
        {
            if(a[i]%j==0)
            {
                isPrime=0;
                break;
            }
        }
        if(isPrime==1)
            printf("%d\n",a[i]);
    }
    return 0;
}

```

(d)

```

#include<stdio.h>
int main()
{
    int i,pos,neg,even,odd;
    int a[25];
    for(i=0;i<25;i++)
        scanf("%d",&a[i]);
    for(i=0;i<25;i++)
    {
        if(a[i]<0)
            neg++;
        else if(a[i]>0)
            pos++;
        else if(a[i]%2==0)
            even++;
        else if(a[i]%2!=0)
            odd++;
    }
    printf("Negatives = %d\n",neg);
    printf("Positives = %d\n",pos);
    printf("Even = %d\n",even);
    printf("Odd = %d\n",odd);
    return 0;
}

```

(e)

```

#include<stdio.h>

```

```

int main()
{
    int i,n;
    printf("Enter size of array = ");
    scanf("%d",&n);
    int a[n];
    for(i=0;i<n;i++)
    {
        scanf("%d",&a[i]);
    }
    for(i=0;i<n;i+=2)
    {
        temp=a[i];
        a[i]=a[i+1];
        a[i+1]=temp;
    }
    for(i=0;i<n;i++)
    {
        printf("%d\n",a[i]);
    }
    return 0;
}

```

[E]

- (a) Error: There will be a comma before variable in printf function.
- (b) Error: There will be a comma before variable in printf function.
- (c) 7
9
11
13
15
- (d) 0
0
0
0
0
- (e) 3 2 15

[F]

- (a) No Error.
- (b) No Error.
- (c) j is not an pointer. It cannot save address.
- (d) No Error.
- (e) i is undeclared.

[G]

- (a) Other data maybe overwritten.
- (b) Address of 0th element or base address.
- (c) Unused elements will be filled with zero's or garbage value.
- (d) Other data maybe overwritten.

- (e) Address of the first element of the array.
- (f) To manipulate parts of an array.
- (g) 0

[H]

- (a) True.
- (b) float *ptr;
- (c) j=*ptr;
- (d) *(s + 2)

[I]

(a)

```
#include<stdio.h>
int main()
{
    int n,i,d=0;
    printf("Enter size of array = ");
    scanf("%d",&n);
    int a[n],b[n];
    for(i=0;i<n;i++)
    {
        scanf("%d",&a[i]);
    }
    for(i=n-1;i>=0;i--)
    {
        b[d]=a[i];
        d++;
    }
    for(i=0;i<n;i++)
    {
        printf("%d\n",b[i]);
    }
    return 0;
}
```

(b)

```
#include<stdio.h>
int main()
{
    int n,i,j;
    printf("Enter size of array = ");
    printf("(size must be even");
    scanf("%d",&n);
    int a[n];
    for(i=0;i<n;i++)
    {
        scanf("%d",&a[i]);
    }
    for(i=0,j=n-1;i<n/2;i++,j--)
    {
        if(a[i]==a[j])
            printf("a[%d] & a[%d] are equal",i,j);
        else
            printf("a[%d] & a[%d] are not equal",i,j);
    }
    return 0;
}
```

(c)

```
#include<stdio.h>
int main()
{
    int i;
    int a[25];
    int *ptr;
    for(i=0;i<25;i++)
    {
        scanf("%d",&a[i]);
    }
    ptr=&a;
    for(i=0;i<25;i++)
    {
        if(*ptr>a[i])
            *ptr=a[i];
    }
    printf("%d",*ptr);
    return 0;
}
```

(d)

```
#include<stdio.h>
int modify(int *);
int main()
{
    int i;
    int a[10]={1,2,3,4,5,6,7,8,9,10};
    modify(a);
    for(i=0;i<10;i++)
    {
        printf("%d\n",a[i]);
    }
    return 0;
}
int modify(int *a)
{
    int i;
    for(i=0;i<10;i++)
    {
        a[i]=a[i]*3;
    }
    return a;
}
```

<https://github.com/hiverkiya/Let-Us-C-Solutions>

MultiDimensional Arrays

14 July 2020 16:13

Chapter-14 Multidimensional Arrays

[A]

(a) [Address of 1st 1d Array] [Garbage Value] 1
(b) 2
4
3
6
8
5
3
5
1
(c) 2 2
4 4
3 3
6 6
8 8
5 5
3 3
5 5
1 1

[B]

(a) missing subscript in initialization.
(b) missing subscript in initialization.

[C]

(a)

```
#include<stdio.h>
int main()
{
int threedimen[3][2][3] = {
{
1, 2, 3,
4, 5, 6
},
{
7, 8, 9,
10, 11, 12
},
{
13, 14, 15,
16, 17, 18
}
};
printf("\nFirst element : %d", threedimen[0][0][0]);
printf("\nLast element : %d", threedimen[2][1][2]);
return 0;
}
```

(b)

```
#include<stdio.h>
int main()
{
int arr[5][5] = {
1,2,3,4,5,
6,7,8,9,10,
```

```

11,12,13,14,15,
16,17,18,19,20,
21,22,23,24,25
};
int i,j,max;
max=arr[0];
for(i=0;i<5;i++)
{
    for(j=0;j<5;j++)
    {
        if(a[i][j]>max)
max=a[i][j];
    }
}
printf("Maximum Number is = %d",max);
return 0;
}

```

(c)

```

#include<stdio.h>
int main()
{
    int i,j;
int a[4][4],trans[4][4];
for(i=0;i<4;i++)
{
    for(j=0;j<4;j++)
    {
        scanf("%d",&a[i][j]);
    }
}
|
for(i=0;i<4;i++)
{
    for(j=0;j<4;j++)
    {
        trans[i][j]=a[j][i];
    }
}
for(i=0;i<4;i++)
{
    for(j=0;j<4;j++)
    {
        printf("%d ",trans[i][j]);
    }
    printf("\n");
}
return 0;
}

```

(d)

```

#include<stdio.h>
#include<conio.h>
#include<Windows.h>
#define FALSE 0
#define TRUE 1
//scan codes for arrow keys.
#define UP 72
#define DOWN 80
#define LEFT 75
#define RIGHT 77
void gotoxy(short, short);
void box(short, short, short, short);
void update(short, short, short, short, int *a);

```

```

int getkey();
int check(int *a);
void action(int key, int a[4][4]);
void swap(int*, int*);
int main()
{
int a[4][4] = {
    1, 4, 15, 7,
    8, 10, 2, 11,
    14, 3, 6, 13,
    12, 9, 5, 32
};
int key, flag = FALSE;
gotoxy(30, 20);
printf("\nArrange the numbes in ascending order.
\nPress any key to start. . .");
_getch();
system("cls");//clrscr(); in TurboC++
box(0, 0, 24, 8);
while (flag == FALSE)//Game ends when flag becomes TRUE
{
    update(0, 0, 24, 8, (int*)a);
    key = getkey();
    action(key, a);// It moves the numbers
    flag = check((int*)a);/*It make flag TRUE on completion of game*/
}
system("cls");
gotoxy(30, 20);
printf("\nYou've done it dude!!!\nExiting. . . ");
_getch();
_getch();
return 0;
}
//It prints the numbers in the boxes.
void update(short x1, short y1, short x2, short y2, int *a)
{
int i, j, num, k = 0;
for (j = y1; j < y2; j++)//row
{
    for (i = x1; i < x2; i++)//col
    {
        if (i > x1 && i < x2 && j > y1 && j < y2)
        {
            if ((i + 3) % 6 == 0 && j % 2 == 1)
            {
                gotoxy(i, j);
                num = a[k];
                if (num == 32)
                    printf(" ");
                else
                    printf("%d", num);
                k++;
            }
        }
    }
}
}
//It draw box.
void box(short x1, short y1, short x2, short y2)
{
int i, j;
for (i = x1; i <= x2; i++) // col

```

```

{
    for (j = y1; j <= y2; j++) //row
    {
        /*====printing corners=====*/
        if (i == x1 && j == y1)
        {
            gotoxy(i, j);
            printf("%c", 218);
        }
        if (i == x2 && j == y1)
        {
            gotoxy(i, j);
            printf("%c", 191);
        }
        if (i == x1 && j == y2)
        {
            gotoxy(i, j);
            printf("%c", 192);
        }
        if (i == x2 && j == y2)
        {
            gotoxy(i, j);
            printf("%c", 217);
        }
        /*=====*/

        if (i < x2 && i > x1)
        {
            if ((i) % 6 == 0) //Vertical division coerners
            {
                if (j == y1)
                {
                    gotoxy(i, j);
                    printf("%c", 194); //T
                }
                if (j == y2)
                {
                    gotoxy(i, j);
                    printf("%c", 193); //Reverse T
                }
                if (j == y1 + 2 || j == y1 + 4 || j == y1 + 6)
                {
                    gotoxy(i, j);
                    printf("%c", 197); //+
                }
            }
            else if (j % 2 == 0) // Horizontal line
            {
                gotoxy(i, j);
                printf("%c", 196);
            }
        }
    }

    if (j > y1 && j < y2)
    {
        if ((j % 2 == 0))
        {
            if (i == x1)
            {
                gotoxy(i, j);
                printf("%c", 195); //Rotate left T
            }
        }
    }
}

```

```

    }
    if (i == x2)
    {
        gotoxy(i, j);
        printf("%c", 180); //Rotate right T
    }
}
else if (i % 6 == 0)
{
    gotoxy(i, j);
    printf("%c", 179); //Vertical lines
}
}
}
}
//gotoxy in compilers other than TurboC++
void gotoxy(short col, short row)
{
    HANDLE h = GetStdHandle(STD_OUTPUT_HANDLE);
    COORD position = { col, row };
    SetConsoleCursorPosition(h, position);
}
int getkey()
{
    int ch;
    ch = _getch();
    if (ch == 0)
        ch = _getch();
    return ch;
}
//It check if the puzzle is solved or not.
int check(int *a)
{
    int i;
    for (i = 0; i < 15; i++)
        if (a[i] > a[i + 1])
            return FALSE;
    return TRUE;
}
//It moves the number.
void action(int key, int a[4][4])
{
    int i, j;
    for (i = 0; i < 4; i++)
        for (j = 0; j < 4; j++)
            if (a[i][j] == 32)
                goto BREAK;
BREAK:
    switch (key)
    {
    case RIGHT:
        if (j - 1 < 0)
            return;
        swap(&a[i][j], &a[i][j - 1]); /*Moves the botton to right*/
        return;
    case DOWN:
        if (i - 1 < 0)
            return;
        swap(&a[i][j], &a[i - 1][j]); /*Moves the botton down*/
        return;
    case LEFT:

```



```

    if (j + 1 > 3)
        return;
    swap(&a[i][j], &a[i][j + 1]);/*Moves the botton left*/
    return;
case UP:
    if (i + 1 > 3)
        return;
    swap(&a[i][j], &a[i + 1][j]);/*Moves the botton up*/
    return;
default:
    return;
}
}
void swap(int *a, int *b)
{
    int temp;
    temp = *a;
    *a = *b;
    *b = temp;
}

```

(e)

```

1. &u      = e. F9C
2. &j      = b. F9E
3. pj      = b. F9E
4. *pj     = c. 35
5. i       = g. unspecified (60)
6. pi      = e. F9C
7. *pi     = g. unspecified (25)
8. (pi+2)  = i. F9E
9. (*pi+2) = g. unspecified (27)
10.*(pi+2) = c. 35

```

(f)

```

1. = j. 2
2. = g. 1
3. = None, it prints the address of a[2][0]th element.
4. = h. 11
5. = b. 18
6. = g. 1
7. = e. 0
8. = k. 5
9. = j. 2
10. = d. 3

```

(g)

```

1. = f. 12
2. = j. 8
3. = m. 6
4. = c. 4
5. = a. 9
6. = i. 1
7. = d. 3
8. = k. 5
9. = h. 7
10. = e. 2

```

(h)

```

#include<stdio.h>
int main()
{
    int i,j,count=0;
    printf("Enter no. of rows & coulumnns = ");

```

```

        scanf("%d",&n);
        int a[n][n];
        for(i=0;i<n;i++)
        {
            for(j=0;j<n;j++)
        {
            scanf("%d",&a[i][j]);
        }
        }
        for(i=0;i<n;i++)
        {
            for(j=0;j<n;j++)
        {
            if(a[i][j]==a[j][i])
            count++;
        }
    }
    int d=n*n;
    if(count==d)
    printf("This is Symmetric Matrix");
    else
    printf("This is not a Symmetric Matrix");
    return 0;
}

```

(i)

```

#include<stdio.h>
int main()
{
    int i,j;
    int a[6][6],b[6][6],sum[6][6];
    for(i=0;i<6;i++)
    {
        for(j=0;j<6;j++)
        {
            scanf("%d",&a[i][j]);
        }
    }
    for(i=0;i<6;i++)
    {
        for(j=0;j<6;j++)
        {
            scanf("%d",&b[i][j]);
        }
    }
    for(i=0;i<6;i++)
    {
        for(j=0;j<6;j++)
        {
            sum[i][j]=a[i][j]+b[i][j];
        }
    }
    for(i=0;i<6;i++)
    {
        for(j=0;j<6;j++)
        {
            printf("%d ",sum[i][j]);
        }
        printf("\n");
    }
    return 0;
}

```

```

(j) #include <stdio.h> #include <stdlib.h> #include<time.h> #include<stddef.h>
#include<string.h> #include<math.h>
void main()
{
    int i,j,k;
    int a[3][3],b[3][3],c[3][3];
    for(i=0;i<3;i++)
    {
        for(j=0;j<3;j++)
        {
            printf("\na[%d][%d]=",i,j);
            scanf("%d",&a[i][j]);
        }
    }
    for(i=0;i<3;i++)
    {
        for(j=0;j<3;j++)
        {
            printf("\nb[%d][%d]=",i,j);
            scanf("%d",&b[i][j]);
        }
    }
    for (i = 0; i<3; i++)
    {
        for (j = 0; j<3; j++)
        {
            c[i][j] = 0;
            for (k = 0; k<3; k++)
            {
                c[i][j] = c[i][j] + a[i][k] * b[k][j];
            }
        }
    }
    for(i=0;i<3;i++)
    {
        for(j=0;j<3;j++)
        {
            printf("%d ",c[i][j]);
        }
    }
    printf("\n");
}
}

```

```

(k)
#include<stdio.h>
#include<conio.h>
void shift(int *base)
{
    int *web, fir, sec, i;
    web = base;
    fir = *base;//saving first value in fir variable
    sec = *(base + 1);//saving second value insec variable
    for (i = 0; i<3; i++)//shifting the values by saving them in next to next
addresses
        *(web + i) = *((base + 2) + i);
    *(web + 3) = fir;
    *(web + 4) = sec;
}
int main()
{

```

```

    int a[5], i;
    printf("Enter 5 numbers : ");
    for (i = 0; i<5; i++)//scanning values
        scanf("%d", &a[i]);
    shift(a);//calling function
    printf("\n\nList after shifting it's rows by two positions.\n\n");
    for (i = 0; i<5; i++)//printing values after shifting
        printf("%d ", a[i]);
    _getch();
    return 0;
}
(1) #include<conio.h> #include<stdio.h>
    int a[20][20],m;
    int determinant(int f[20][20],int a);
    int main()
    {
    int i,j; printf("\n\nEnter order of matrix : "); scanf("%d",
    &m); printf("\nEnter
    the elements of matrix\n"); for(i=1;i<=m;i++) { for(j=1;j<=m;j++) {
    printf("a[%d][%d] = ",i,j); scanf("%d",&a[i][j]); } } printf("\n\n-----
    Matrix A is -----\\n");
    for(i=1;i<=m;i++) { printf("\\n"); for(j=1;j<=m;j++) {
    printf("\\t%d \\t",a[i][j]); } } printf("\\n \\n"); printf("\\n Determinant of
    Matrix A is %d .",determinant(a,m)); getch(); }
    int determinant(int f[20][20],int x)
    {
        int pr,c[20],d=0,b[20][20],j,p,q,t;
        if(x==2)
{ d=0; d=(f[1][1]*f[2][2])-(f[1][2]*f[2][1]); return(d); } else {
for(j=1;j<=x;j++) {
    int r=1,s=1; for(p=1;p<=x;p++) { for(q=1;q<=x;q++) { if(p!=1&&q!=j) { b[r]
[s]=f[p][q];
s++; if(s>x-1) { r++; s=1; } } } } for(t=1,pr=1;t<=(1
+j);t++) pr=(-1)*pr; c[j]=pr*determinant(b,x-1);
} for(j=1,d=0;j<=x;j++) { d=d+(f[1][j]*c[j]); } return(d); } }
(m)
#include<stdio.h>
#include<conio.h>
#include<math.h>
int main()
{
int a[15] = { -6,-12,8,13,11,6,7,2,-6,-9,-10,11,10,9,2 },
i, sum = 0;
float min, hell = 0, sd, n = 0;
printf("Your data is :");
for (i = 0; i<15; i++)//sum of the data
{
    printf(" %d,", a[i]);
    sum = sum + a[i];
    n++;
}
min = sum / n;//calculating mean of the data.
for (i = 0; i<15; i++)//finding standard deviation of the data
    hell = hell + pow((a[i] - min), 2);
sd = (sqrt(hell)) / n;
printf("\\n\\nMean of the data = %f\\n\\nStandard deviation of data = %
f", min, sd);
_getch();
return 0;
}
(o)
#include<stdio.h>

```

```

#include<conio.h>
#include<math.h>
int main()
{
float x[11][2] = {
    34.22, 102.43,
    39.87, 100.93,
    41.85, 97.43,
    43.23, 97.81,
    40.06, 98.32,
    53.29, 98.32,
    53.29, 100.07,
    49.12, 91.59,
    40.71, 94.85,
    55.15, 94.65,
},
    xysum = 0, xsum = 0, ysum = 0, x2sum = 0, y2sum = 0, n = 11, r;
int i;
for (i = 0; i<11; i++)
{
    xsum = xsum + x[i][0];
    ysum = ysum + x[i][1];
    xysum = xysum + x[i][0] * x[i][1];
    x2sum = x2sum + x[i][0] * x[i][0];
    y2sum = y2sum + x[i][1] * x[i][1];
}
r = (xysum - xsum*ysum) / (sqrt((n*x2sum - xsum*xsum)*(n*y2sum - ysum*ysum
)));
printf("Coefficient of correlation (r) = %f", r);
_getch();
return 0;
}

```

(p)

```

#include<stdio.h>
#include<conio.h>
#define n 10
int main()
{
double set[n][2] = {
    3.0, 1.5,
    4.5, 2.0,
    5.5, 3.5,
    6.5, 5.0,
    7.5, 6.0,
    8.5, 7.5,
    8.0, 9.0,
    9.0, 10.5,
    9.5, 12.0,
    10.0, 14.0
};
double help[n], sx = 0, sy = 0, sxy = 0, sx2 = 0, a, b, xbar, ybar;
int i;
for (i = 0; i < n; i++)
{
    sx += set[i][0];
    sy += set[i][1];
    sxy += set[i][0] * set[i][1];
    sx2 += set[i][0] * set[i][0];
}
xbar = sx / n;
ybar = sy / n;
b = (n*sxy - sx*sy) / (n*sx2 - sx*sx);

```

```

a = ybar - b*xbar;
printf("\nRequired equation is\n\ny = %.2lf + %.2lfx", a, b);
_getch();
return 0;
}

```

(q)

```

#include<stdio.h>
#include<conio.h>
#include<math.h>
int main()
{
float x[10], y[10], dis = 0;
int i, j;
printf("Enter the coordinates of 10 points ~ \n\n");
for (i = 0; i<10; i++)
    scanf("%f%f", &x[i], &y[i]);
for (i = 0; i<10; i++)
    dis = dis + sqrt(pow((x[i + 1] - x[i]), 2) + pow((y[i + 1] - y[i]), 2));
printf("The total distance between first and last point is %f", dis);
_getch();
return 0;
}

```

(n)

```

#include<stdio.h>
#include<conio.h>
#include<math.h>
int main()
{
float a[6][4] = {
    // Plot No.    a        b        angle
    1,    137.4,    80.9, 0.78,
    2,    155.2,    92.62, 0.89,
    3,    149.3,    97.93, 0.89,
    4,    160.0,    100.25, 1.35,
    5,    155.6,    68.95, 1.25,
    6,    149.7,    120.0, 1.75,
},
ar[6], //areas
max;
int i, x = 0; //plot no.
for (i = 0; i<6; i++)
{
    ar[i] = (0.5)*a[i][1] * a[i][2] * sin(a[i][3]);
    if (max<ar[i])
    {
        max = ar[i];
        x = a[i][0];
    }
}
printf("\n\n\nAreas of the triangles are ~ ");
for (i = 0; i<6; i++)
    printf("\n%f : %2f,", a[i][0], ar[i]);
printf("\n\n\nNumber %d. triangle has maximum area of %2f.", x, max);
_getch();
return 0;
}

```

(r)

```

#include<stdio.h>
#include<conio.h>
#define MAX 10
void lAdd(int *l, int *q, int *r, int num);
void rAdd(int *l, int *q, int *r, int num);

```

```

void show(int *q, int *rside);
int rFetch(int *l, int *q, int *r);
int lFetch(int *l, int *q, int *r);
int main()
{
    int que[MAX];
    int left, right;
    left = right = -1;
    rAdd(&left, que, &right, 10);
    rAdd(&left, que, &right, 20);
    rAdd(&left, que, &right, 30);
    show(que, &right);
    lAdd(&left, que, &right, 5);
    show(que, &right);
    lFetch(&left, que, &right);
    show(que, &right);
    rFetch(&left, que, &right);
    show(que, &right);
    _getch();
    return 0;
}
/*Insertion from right*/
void rAdd(int *lside, int *q, int *rside, int num)
{
    if (*rside == MAX - 1)
    {
        printf("\nDeque is full, no more insertion is possible.\n");
        return;
    }
    if (*rside == -1)
        *rside = 0;
    else
        (*rside)++;
    q[*rside] = num;
}
/*Insertion from left*/
void lAdd(int *lside, int *q, int *rside, int num)
{
    int i;
    if (*rside == MAX - 1)
    {
        printf("\nDeque is full, no more insertion is possible.\n");
        return;
    }
    for (i = *rside; i >= 0; i--)
        q[i+1] = q[i];
    q[0] = num;
    if (*lside == -1)
        *lside = 0;
    (*rside)++;
}
/*Displays the list*/
void show(int *q, int *rside)
{
    int i;
    printf("\n\nList\n");
    for (i = 0; i <= *rside; i++)
        printf("%d\t", q[i]);
    printf("\n\n");
}
/*Retrieval from left*/
int lFetch(int *lside, int *q, int *rside)

```

```

{
int item = q[0], i;
if (*rside == -1)
{
    printf("\nList is empty.\n");
    return NULL;
}
for (i = 0; i < *rside; i++)
    q[i] = q[i + 1];
(*rside)--;
return item;
}
/*Retrieval from right*/
int rFetch(int *lside, int *q, int *rside)
{
if (*rside == -1)
{
    printf("\nList is empty.\n");
    return NULL;
}
int item = q[*rside];
(*rside)--;
return item;
}
(s)
#include<stdio.h>
#include<conio.h>
static int error = 0;
void grd(int *a)//checking the 3x3 grids
{
int i, j, x, p, sum = 0;
for (p = 0; p <= 54; p += 27)/*Moving down the grid*/
{
    for (x = 0; x <= 6; x += 3)/*Moving right of the grid*/
    {
        sum = 0;
        for (i = 0; i <= 18; i += 9)//Moving down the column in a grid
            for (j = 0; j <= 2; j++)//Moving right of row in a grid
                sum += *(a + i + j + x + p);
        if (sum != 45)
        {
            error += 1;
            return;
        }
    }
}
}
void row(int *a)//checking the rows
{
int i, j, sum = 0;
for (i = 0; i <= 72; i += 9)/*Moving down the row*/
{
    sum = 0;//reinitialisation of sum
    for (j = 0; j <= 8; j++)//Moving right of the row
        sum += *(a + i + j);
    if (sum != 45)
    {
        error += 1;//if there's error
        return;
    }
}
}
}

```



```

void col(int *a)//checking the columns
{
int i, j, sum = 0;
for (i = 0; i <= 8; i++)//Moving right of the column
{
    sum = 0;//reinitialisation of sum
    for (j = 0; j <= 72; j += 9)//Moving down of the column
        sum += *(a + i + j);
    if (sum != 45)
    {
        error += 1;//if ther's error
        return;
    }
}
}
int main()
{
int gam[9][9] = {
    5,3,4,6,7,8,9,1,2,
    6,7,2,1,9,5,3,4,8,
    1,9,8,3,4,2,5,6,7,
    8,5,9,7,6,1,4,2,3,
    4,2,6,8,5,3,7,9,1,
    7,1,3,9,2,4,8,5,6,
    9,6,1,5,3,7,2,8,4,
    2,8,7,4,1,9,6,3,5,
    3,4,5,2,8,6,1,7,9
};
grd(gam[0]);
row(gam[0]);
col(gam[0]);
if (error)
    printf("\n\nThe Solution of your sudoku is wrong.\n");
else
    printf("\n\nSolution of you sudoku is right.\n");
_getch();
return 0;
}

```

<https://github.com/hiverkiya/Let-Us-C-Solutions>

Strings

14 July 2020 16:13

[Chapter 15: Strings] Solutions

[A]

(a)

A
A

(b)

t organised! learn C!!
Get organised! learn C!!
Get organised! learn C!!
t

(c)

N N
N N
o o
o o

t t
t t
w w
w w
o o
o o

v v
v v
i i
i i
r r
r r
u u
u u
s s
s s
e e
e e
s s
s s

w w
w w
o o
o o
r r
r r
k k
k k

s s
s s
i i
i i
m m

```
m m
i i
i i
l l
l l
a a
a a
r r
r r
l l
l l
y y
y y
```

(d)

Garbage or error, that t is not initialized.

(e)

```
Hello[garbage characters because NULL character is missing]
Hello
```

(f)

```
Morning
```

(g) e

(h) 1 2 4

[B]

(a)

Error: As the str3 is declared but not defined, so it is only void pointer, so without allocating memory to it, we cannot store anything to it.

(b) No Error

(c) Error: In arr++ statement. As you can see that arr[8] is initialized at the time of declaration. So in that case we cannot assign anything to arr, and statement arr++ is execute as arr = arr + 1, which is illegal.

[C]

(a) string, character

(b) NULL, '\0'

(c) 9 characters, and 10th character space is for \0 character.

(d) contiguous.

[D]

(a)

gets() is more appropriate for reading a multi-word string.

(b)

```
str1 : Alice
str2 : in
str3 : wonder
str4 : land
```

(c)

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
int main()
{
    char sen[100];
    int i;
    printf("Enter your sentence. (Less than 100 chracters)\n\n");
    gets_s(sen);
    printf("After how muany characters you want to extract the line : ");
    scanf("%d", &i);
    i--;
    if (i<0)
        i = 0;
    printf("%s\t", &sen[i]);
    _getch();
    return 0;
}
```

(d)

```
#include<stdio.h>
#include<conio.h>
int main()
{
    char str[100];
    int i, num, fin = 0;
    printf("Enter a string of numbers : ");
    gets_s(str);
    for (i = 0; str[i] != '\0'; i++)/*Run until string is not terminated.*/
    {
        num = str[i] - 48;/
*Converting the character number into integer number.*/
        fin = fin * 10 + num;/
*Combining the individual numbers to form a single integers*/
    }
    printf("\nNumber : %d", fin);
    _getch();
    return 0;
}
```

(e)

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
int main()
{
    char x[50] = "a", z[50];
    char y[50] = "b";
    int i;
    for (i = 0; i <= 5; i++)
    {
        printf("%s\t", x);/*printing x
        strcpy(z, x);/*copying x into z*/
        strcpy(x, y);/*copying y into x*/
        strcat(y, z);/*adding (concatinating) z in y*/
    }
    _getch();
    return 0;
}
```

```
}
```

(f)

```
#include<stdio.h>
#include<conio.h>
int main()
{
    char isbn[15];
    int i, sum = 0;
    printf("\nEnter 10 digit ISBN number : ");
    gets_s(isbn);
    for (i = 0; i <= 9; i++)
    {
        isbn[i] -= 48; /*Converting characters into numerals*/
        sum = sum + ((i + 1)*isbn[i]);/
/*checking the condition of the ISBN validity*/
    }
    if (sum % 11)/*If not divisble by 11*/
        puts("\nISBN number is wrong.");
    else
        puts("\nISBN number is valid.");
    _getch();
    return 0;
}
```

(g)

```
#include<stdio.h>
#include<conio.h>
int main()
{
    char num[20];
    int i, sum = 0;
    printf("\nEnter the 16 digit credit card number : ");
    scanf("%s", num);
    for (i = 0; i <= 15; i++)//Traversing all numbers
    {
        num[i] -= 48;//converting each character in numeral
        if ((i % 2))//if the number is on right so it will directly get summed
            sum = sum + num[i];
        else//if number is on left, so it will first get doubled
        {
            num[i] *= 2;
            if (num[i] >= 10)//if number is greater then or equal to 10 so it will sub
traced from 9
            num[i] -= 9;
            sum = sum + num[i]; //summing number
        }
    }
    if (!(sum % 10))//if sum is divisble by 10 so number is valid
        printf("\nNumber is valid.");
    else
        printf("\nNumber is not valid.");
    _getch();
    return 0;
}
```

<https://github.com/hiverkiya/Let-Us-C-Solutions>

Handling Multiple Strings

14 July 2020 16:13

[Chapter 16: Handling Multiple Strings] Solutions

[A]

(a)

58

(b)

An array of strings can never collect string from keyboard directly, it can only be initialized at the time of declaration. But still we can store strings from the keyboard in the array of characters indirectly, by storing the string in the simple string variable and then allocate memory to the pointer of appropriate size and then copy the string from that simple string variable to the array of pointer.

[B]

(a)

```
#include<stdio.h>
#include<conio.h>
#include<malloc.h>
#include<string.h>
#define LINE 6/*If the number of lines are increased
so no need to change anything in the program just change
LINE macro here.*/
#pragma warning (disable : C4267)
void ser_rep(char **str, char *old, char *news)
{
    int i, j, k, m, c;
    j = k = m = i = c = 0;
    char ans[10][100], *p;

    for (i = 0; i <= LINE - 1; i++)
    {
        m = 0;
        c = 0;
        /*This loop for traversing the current line character by character.*/
        for (j = 0; str[i][j] != '\0';)
        {
            if (str[i][c] == old[k])/*If the character matches*/
            {
                k++;
                c++;
                if (old[k] == '\0')/*If the whole word is found.*/
                {
                    /*This loop is replacing the word character by character.*/
                    for (k = 0; news[k] != '\0'; k++, m++)
                        ans[i][m] = news[k];
                    j = c;
                    k = 0;
                }
            }
        }
    }
}
```

```

        else/
/*If the character/word is not matched, so as it is string is copied.*/
        {
            ans[i][m] = str[i][j];
            m++;
            j++;
            c = j;
            k = 0;
        }
    }
    ans[i][m] = '\0'; /*Terminate every line by NULL character*/
    j = k = m = 0; /*Reinitialization for next line.*/
}
for (i = 0; i <= LINE - 1; i++)
{
    p = (char*)malloc(strlen(ans[i]) + 1);
    /*Memory needed is 1 more for null character.
    str[i] = ans[i]; This doesn't works becuae left side is pointer and
    right side is simple
    char variable therefore a news pointer p is taken and copied the con
    tent of ans[i] and
    assign to the str[i].*/
    strcpy(p, ans[i]);
    str[i] = p;
}
}
int main()
{
    char *str[] = {
        "We will teach you how to...",
        "Move a mountain",
        "Level a building",
        "Erase the past",
        "Make a million",
        "...all through C!"
    };
    char str1[10], str2[10];
    puts("Enter the word you want to replace.");
    scanf("%s", str1);
    puts("Enter the word you want to put.");
    scanf("%s", str2);
    ser_rep(str, str1, str2);
    for (int i = 0; i <= LINE - 1; i++)
        puts(str[i]);
    _getch();
    return 0;
}

```

(b)

```

#include<stdio.h>
#include<conio.h>
#include<string.h>
#include<malloc.h>
#include<Windows.h>
void swap(char *a, char *b)
{
    char temp[20];
    strcpy(temp, a);
    strcpy(a, b);
    strcpy(b, temp);
}
void sort_names(char** name_list, int tot_names)
{

```

```

int i, j, k = 0;
for (i = 0; i < tot_names; i++)
{
    for (j = i + 1; j < tot_names; j++)
    {
        k = 0;
        while (name_list[i][k] == name_list[j][k])
            k++;
        if (name_list[i][k] > name_list[j][k])
            swap(name_list[j], name_list[i]);
    }
}
}
int main()
{
    char *nam_list[10], *p, name[20], ans = 'y';
    int i = 0, j = 0, a, b;
    while (ans == 'y')
    {
        printf("\nEnter the name : ");
        scanf("%s", name);
        p = (char*)malloc(strlen(name));
        strcpy(p, name);
        nam_list[i] = p;
        i++;
        while (getchar() != '\n');
        printf("\nWant to enter another name (y/n) : ");
        scanf("%c", &ans);
    }
    puts("\n");
    sort_names(nam_list, i);
    system("cls");
    puts("\n\t\t\tNames in sorted order are follows.\n\n");
    for (j = 0; j <= i - 1; j++)
        puts(nam_list[j]);
    _getch();
    return 0;
}

```

(c)

```

#include<stdio.h>
#include<conio.h>
#include<string.h>
#include<malloc.h>
#include<Windows.h>
int main()
{
    char *s[] = {
        "To err is human",
        "But to really mess up things...",
        "One needs to know C!!"
    };
};

```

/*As the given strings are stored in the ROM (initialized direct in the pointer saved them in ROM), so they cannot be write only Read operation can be done. So we just replace the pointer saved in the array of pointer to string by a new pointer in which we make changes.*/

```

char *p; /*Pointer in which we will make changes*/
for (int i = 0; i <= 2; i++)
{
    p = (char*)malloc(sizeof(strlen(s[i]) + 1)); /*Allocating memory*/
    strcpy(p, s[i]); /*Copying content to be changed from the pointer

```



```

        initialized to the new pointer*/
        _strrev(p);/*Reverse Operation taking place*/
        s[i] = p; /*Saving the changes in the array of pointers to string.*/
        puts(s[i]);
    }
    _getch();
    return 0;
}
(d)
#include<stdio.h>
#include<conio.h>
#include<Windows.h>
#define VS /*Uncomment this if using Visual Studio*/
// #define TC /*Uncomment this if using Turbo C++*/
#define LEAP (!(yr%4))/*Cheking for leap year*/
int main()
{
    int j = 1, a = 7, i, x = 1, nod;
    int year, yr, month, m, mon, curr, next = 0;

    /*yr is for traversing the year, andn year is the
    year given by the user. mon is for traversing the
    months and month is the month given by user. curr
    contains the starting day of the current month
    and next conatains the starting day of the next month.*/

    printf("\nEnter the year : ");
    scanf("%d", &year);
    printf("\nEnter the month : ");
    scanf("%d", &month);

    #ifdef VS
    system("cls");/*clearing screen in visula studio.*/
    #endif
    #ifdef TC
    clrscr();
    #endif

    for (yr = 1900; yr <= year; yr++)/*Traversing years*/
    {
        if (yr < year)
            m = 12; /*It will traverse the all months if
            the year is not what user entered*/
        else
            m = month; /*It will traverse till the month user
            want if the wanted year is encountered.*/

        for (mon = 1; mon <= m; mon++)//traversing all months of the year
        {
            if (mon == 2)/*If the feb is encountered*/
            {
                if (LEAP)
                    nod = 29;
                else
                    nod = 28;
            }
            else if (mon == 4 || mon == 6 || mon == 9 || mon == 11)
                /*Months with 30 days*/
                nod = 30;
            else/*Months with 31 days*/
                nod = 31;
            int i = 1;

```

```

        curr = next; /*Current month starts from this day*/
        while (i <= nod)
        /*This will find the day from which next month will start*/
        {
            next++;
            i++;
            if (next == 7)
                next = 0;
        }
    }
}

//x is dates
j = 1, a = 7, i, x = 1;
if (month == 1)
    printf("\n\t\tJanuary");
if (month == 2)
    printf("\n\t\tFebruary");
if (month == 3)
    printf("\n\t\tMarch");
if (month == 4)
    printf("\n\t\tApril");
if (month == 5)
    printf("\n\t\tMay");
if (month == 6)
    printf("\n\t\tJune");
if (month == 7)
    printf("\n\t\tJuly");
if (month == 8)
    printf("\n\t\tAugust");
if (month == 9)
    printf("\n\t\tSeptember");
if (month == 10)
    printf("\n\t\tOctober");
if (month == 11)
    printf("\n\t\tNovember");
if (month == 12)
    printf("\n\t\tDecember");
printf("\t %d\n\n", year);
printf("\n\n  Sun   Mon   Tue   Wed   Thu   Fri   Sat\n\n");
for (i = 1; x <= nod; i++)
{
    for (j = 1; j <= 7; j++, curr--)
    {
        if (x>9)//maintaining the structure of the calender
            printf("   ");
        else
            printf("   ");
        if (curr>0)//spacing for start the calender from the given day
            printf(" ");
        else//printing of dates begin.
        {
            printf("%d", x);
            if (x <= nod)
                x++;
            if (x>nod)
                break;
        }
    }
    printf("\n\n");//next week
    if (x>nod)
        break;
}

```

```

    }
    _getch();
    return 0;
}
(e)
#include<stdio.h>
#include<conio.h>
#include<dos.h>
/*****
/* function to tackle arrow keys */
*****/
getkey() {
    union REGS i,o;
    while(!kbhit());
    i.h.ah=0;
    int86 (22,&i,&o);
    return (o.h.ah);
}
void main() {

    int x,y,i,lastday,key;
    int month,year,a;
    void box();
    clrscr();
    printf("Enter month: ");
    scanf("%d",&month);
    printf("\n\nEnter year: ");
    scanf("%d",&year);
    /*****
    /* starting the program with a condition */
    *****/

    if(month<=12 && month>=1 && year>=1900 && year<=2045) {
    do {

        /* if up arrow key is hit */

        if(key==72) {
            if(year+1 > 2045) {
            }
            else {
                year=year+1; /* increment of year */
            }
        }

        /* if down arrow key is hit */
        if(key==80) {
            if(year-1 < 1900) {
            }
            else {
                year=year-1; /* decrement of year */
            }
        }

        /* if left arrow key is hit */
        if(key==75) {
            if(month-1 < 1){
            }
            else {
                month=month-1; /* decrement of month */
            }
        }
    }
}

```

```

/* if right arrow key is hit */
if(key==77) {
if(month+1 > 12){
}
else {
month=month+1; /* increment of month */
}
}
x=49,y=9,i=1; /* calender printing objects */

x = dayfinder(month,year); /* calculating first day of the month */
lastday = totaldays(month,year); /*
* calculating total days of the month*/

clrscr();
box(month,year); /* drawing boxes and headings of calender */

/*****
/* printing the calender */
*****/

while(i<=lastday) {
gotoxy(x,y);
printf("%2d",i);
i++;
x+=5;
if(x>52) { /* if the position of 7 days is covered, again print from
beginning from a new line */
x=22;
y+=2;
}
}
gotoxy(1,1); /* moving cursor away from calender */
key=getkey(); /* taking the arrow key input */
} while(key==72 || key==75 || key==77 || key==80);
}
else
printf("Error! invalid input\n");

getch();
}
/***** main ends *****/

/*****
/* function to find first day of the given month and year */
*****/
int dayfinder(int month, int year)
{
int a,day=1;
/* this is a general purpose formula to calculate first day */
a=(14-month)/12;
year=year-a;
month=month+12*a-2;
day=(day+year+(year/4)-(year/100)+(year/400)+((31*month)/12)) % 7;

/* determining the position to print the first day in the calender */
if(day==0)
day=22;
else if(day==1)
day=27;

```

```

else if(day==2)
day=32;
else if(day==3)
day=37;

else if(day==4)
day=42;
else if(day==5)
day=47;
else if(day==6)
day=52;
return (day); /* return the position */
}
/*****
/* function to draw the boxes, headings of the calender */
*****/

void box(int m,int y) {

int i,j,k,l;
/*****/
/* inner box */
/*****/
/* corners of inner box */
gotoxy(20,3);
printf("%c",218);

gotoxy(55,3);
printf("%c",191);
gotoxy(55,21);
printf("%c",217);
gotoxy(20,21);
printf("%c",192);
/* boundries of inner box */
for(j=4;j<=20;j++) {
gotoxy(20,j);
printf("%c",179);

gotoxy(55,j);
printf("%c",179);
}

for(i=21;i<=54;i++) {
gotoxy(i,3);
printf("%c",196);
gotoxy(i,21);
printf("%c",196);
}
/*****/
/* outer box */
/*****/
/* corners of outer box */
gotoxy(17,1);
printf("%c",218);
gotoxy(17,23);
printf("%c",192);
gotoxy(58,1);
printf("%c",191);
gotoxy(58,23);
printf("%c",217);
/* boundries of outer box */
for(k=2;k<=22;k++) {

```

```

gotoxy(17,k);
printf("%c",179);
gotoxy(58,k);
printf("%c",179);
}

for(l=18;l<=57;l++) {
gotoxy(l,1);
printf("%c",196);
gotoxy(l,23);
printf("%c",196);
}
/*****
/* writing heading on appropriate positions */
*****/

gotoxy(22,6);
printf("Sun");

gotoxy(27,6);
printf("Mon");
gotoxy(32,6);
printf("Tue");
gotoxy(37,6);
printf("Wed");
gotoxy(42,6);
printf("Thu");
gotoxy(47,6);
printf("Fri");
gotoxy(52,6);
printf("Sat");

gotoxy(32,4);
if(m==1)
printf("January %d",y);
if(m==2)
printf("February %d",y);
if(m==3)
printf("March %d",y);
if(m==4)
printf("April %d",y);
if(m==5)
printf("May %d",y);
if(m==6)
printf("June %d",y);
if(m==7)
printf("July %d",y);
if(m==8)
printf("August %d",y);
if(m==9)
printf("September %d",y);
if(m==10)
printf("October %d",y);
if(m==11)
printf("November %d",y);
if(m==12)
printf("December %d",y);

/*****
/* printing instructions */
*****/
gotoxy(60,16);

```

```

printf("%c : Next year",30);
gotoxy(60,18);
printf("%c : Previous year",31);
gotoxy(60,20);
printf("%c : Next month",16);
gotoxy(60,22);
printf("%c : Previous month",17);

}
/*****
/* function to determine total days of given month */
*****/
int totaldays(int m,int y) {
int days;
/* for january */
if(m==1)
days=31;
/* for february */
if(m==2) {
if(y%4==0)
days=29;
else
days=28;
}
/* for march */
if(m==3)
days=31;
/* for april */
if(m==4)
days=30;
/* for may */
if(m==5)
days=31;
/* for june */
if(m==6)
days=30;

/* for july */
if(m==7)
days=31;
/* for august */
if(m==8)
days=31;
/* for september */
if(m==9)
days=30;
/* for october */
if(m==10)
days=31;
/* for november */
if(m==11)
days=30;

/* for december */
if(m==12)
days=31;

return days;
}
(f)
#include<stdio.h>
#include<conio.h>

```

```

#include<Windows.h>
//#define DEVC /*Uncomment this if using DevC++*/
#define VS /*Uncomment this if using Visual Studio*/
//#define TC /*Uncomment this if using Turbo C++*/
#define LEAP (!(yr%4))/*Checing for leap year*/
#define UpArrowKey 72
#define DownArrowKey 80
#define LeftArrowKey 75
#define RightArrowKey 77
#define EscapeKey 27
int getkey()
{
    int ch;
    ch = _getch();
    if (ch == 0)
    {
        ch = _getch();
        return ch;
    }
    return ch;
}
void short_cal(int year, int month)
{
    system("cls");
    int j = 1, a = 7, i, x = 1, nod;
    int yr, m, mon, curr, next = 0;
    for (yr = 1900; yr <= year; yr++)/*Traversing years*/
    {
        if (yr < year)
            m = 12; /
/*It will traverse the all months if the year is not what user entered*/
        else
            m = month;/
/*It will traverse till the month user want if the wanted year is encountered.
*/
        for (mon = 1; mon <= m; mon++)//traversing all months of the year
        {
            if (mon == 2)/*If the feb is encountered*/
            {
                if (LEAP)
                    nod = 29;
                else
                    nod = 28;
            }
            else if (mon == 4 || mon == 6 || mon == 9 || mon == 11)/
/*Months with 30 days*/
                nod = 30;
            else/*Months with 31 days*/
                nod = 31;
            int i = 1;
            curr = next;/*Current month starts from this day*/
            while (i <= nod)/
/*This will find the day from which next month will start*/
            {
                next++;
                i++;
                if (next == 7)
                    next = 0;
            }
        }
    }
}

```



```

//x is dates
j = 1, a = 7, i, x = 1;
printf("\n\n\n\t\t\t\t\t");
if (month == 1)
    printf("January");
if (month == 2)
    printf("Febraury");
if (month == 3)
    printf("March");
if (month == 4)
    printf("April");
if (month == 5)
    printf("May");
if (month == 6)
    printf("June");
if (month == 7)
    printf("July");
if (month == 8)
    printf("August");
if (month == 9)
    printf("September");
if (month == 10)
    printf("October");
if (month == 11)
    printf("November");
if (month == 12)
    printf("December");
printf("\t %d\n\n", year);
printf("\n\n\n\t\t\t\t\t Sun   Mon   Tue   Wed   Thu   Fri   Sat\n\n");
for (i = 1; x <= nod; i++)
{
    printf("\t\t\t\t\t");
    for (j = 1; j <= 7; j++, curr--)
    {
        if (x>9)//maintaining the structure of the calender
            printf("    ");
        else
            printf("    ");
        if (curr>0)//spacing for start the calender from the given day
            printf(" ");
        else//printing of dates begin.
        {
            printf("%d", x);
            if (x <= nod)
                x++;
            if (x>nod)
                break;
        }
    }
    printf("\n\n");//next week
    if (x>nod)
        break;
}
}
int main()
{
    int year, month, key = 0;

    /*yr is for traversing the year, andn year is the year given by the user.
    mon is for traversing
    the months and month is the month given by user.
    curr contains the starting day of the current

```

month and next contains the starting day of the next month.*/

```
printf("\nEnter the year : ");
scanf("%d", &year);
printf("\nEnter the month : ");
scanf("%d", &month);

#ifdef DEVC
    system("cls");/*clearing screen in DevC++.*/
#endif
#ifdef VS
    system("cls");/*clearing screen in Visual Studio.*/
#endif
#ifdef TC
    clrscr();/*clearing screen in TurboC++.*/
#endif
short_cal(year, month);
while (key != 27)
{
    //getchar();
    //while (getkey != '\0');
    fflush(stdin);
    //while (getch() != NULL);
    //getkey();
    key = getkey();
    if (key == EscapeKey)
        return 0;
    if (key == UpArrowKey)
        short_cal(++year, month);
    if (key == DownArrowKey)
        short_cal(--year, month);
    if (key == RightArrowKey)
    {
        if (month == 12)
        {
            month = 0;
            year++;
        }
        short_cal(year, ++month);
    }
    if (key == LeftArrowKey)
    {
        if (month == 1)
        {
            month = 13;
            year--;
        }
        short_cal(year, --month);
    }
}
_getch();
return 0;
}
```

(g)

```
#include<stdio.h>
#include<conio.h>
#include<Windows.h>
#define Vowel line[i] == 'A' || line[i] == 'a' || line[i] == 'E' || line[i]
] == 'e' \
    || line[i] == 'I' || line[i] == 'i' || line[i] == 'O' || line[i] == 'o' |
| \
    line[i] == 'U' || line[i] == 'u'
```

```

void del_vow(char *line)
{
    int i, j;
    for (i = 0; line[i] != '\0'; i++)
        if (Vowel)
            for (j = i; line[j] != '\0'; j++)
                line[j] = line[j + 1];
}

int main()
{
    char line[80];
    puts("Enter the line");
    gets_s(line);
    del_vow(line);
    printf("\nLine without vowels\n");
    puts(line);
    _getch();
    return 0;
}

```

(h)

```

#include<stdio.h>
#include<conio.h>
#include<string.h>
#define Space 32
#define (line[i] == 't' || line[i] == 'T') && (line[i + 1] == 'h' || \
line[i + 1] == 'H') && (line[i + 2] == 'E' || line[i + 2] == 'e') && \
(line[i + 3] == Space || line[i + 3] == '\0'
void del_the(char *line)
{
    int i, j;
    for (i = 0; line[i] != '\0'; i++)
        if (/*if The is encounter*/
            for (j = i; line[j] != '\0'; j++)
                line[j] = line[j + 4];/

```

The word is skipped and other words are saved in the string/

```

}
int main()
{
    char line[80];
    puts("Enter the line");
    gets_s(line);
    del_the(line);
    puts("\nAfter removing all the words 'The'.\n\n");
    puts(line);
    _getch();
    return 0;
}

```

(i)

```

#include<stdio.h>
#include<conio.h>
#include<string.h>
#include<Windows.h>
#include<malloc.h>
#define Space 32/*ASCII v*/
char* last_name(char *line)/*It will return char pointer*/
{
    char temp[20], *p;
    int i, j, l = 0;
    for (i = j = 0; line[i] != '\0'; i++)
    {
        if (line[i] == Space)

```

```

/
*If Space is encountered so the first letter of the word will be saved in the temp*/
{
    temp[l] = line[j];
    l++;
    temp[l] = Space; /*after first letter we are giving a space*/
    l++;
    j = i + 1;
}
}
if (line[i] == '\0')
{
    for (; line[j] != '\0'; j++, l++)
        temp[l] = line[j];
}
/*As if a pointer string cannot be assigned to a simple char variable so,
a pointer
of char is made and then assignment is performed*/
p = (char*)malloc(sizeof(strlen(temp) + 1)); /*p getting size*/
strcpy(p, temp); /*adding content on p*/
return p;
/
*assigning p to the name[i] in main function as this function returns char pointer*/
}
int main()
{
    char *name[10];
    int i = 0, j;
    char ans = 'y', *p, naam[30];
    while (ans == 'y')
    {
        puts("\nEnter the full name : ");
        gets_s(naam);
        p = (char*)malloc(sizeof(strlen(naam) + 1));
        strcpy(p, naam);
        name[i] = p;
        name[i] = last_name(name[i]);
        printf("\nWant to enter another name (y/n) : ");
        scanf("%c", &ans);
        i++;
        while (getchar() != '\n');
        /*So that we can get the next name, otherwise
        \n will be saved in the buffer. (fflush(stdin) doesn't work)*/
        if (i > 9) /*Names cannot be more than 10*/
            break;
    }
    if (i >= 10) /*If trying to enter names more than 10*/
        puts("\nNo more names can be entered");
    system("cls");
    puts("\n\t\t\tName in the given format.");
    for (j = 0; j < i; j++)
        puts(name[j]);
    _getch();
    return 0;
}
(j)
#include<stdio.h>
#include<conio.h>
#define VowelisOccur str[i]=='a' || str[i]=='A' || str[i]=='e' || str[i]=='E' \
|| str[i]=='i' || str[i]=='I' || str[i]=='o' || str[i]=='O' || str[i] == 'u' \

```

```

|| str[i] == 'U'
int find_vows(char * str)
{
int i, count = 0;
printf("\nVowels in successions : ");
for (i = 0; str[i] != '\0'; i++)
{
if (VowelisOccur)/*If first vowel is occur*/
{
i++;/*To check next vowel*/
if (VowelisOccur)/*If second vowel after first if occur*/
{
printf("%c%c ", str[i - 1], str[i]);/*Printing two simultaneous vowels*/
count++;/*Counting Two simultaneous vowels*/
}
}
}
return count;
}
int main()
{
char str[100], tot_vow;
printf("Enter the string : ");
gets_s(str);
tot_vow = find_vows(str);
printf("\n\nTotal number of vowels in successions are : %d\n", tot_vow);
_getch();
return 0;
}

```

(k)

```

#include<stdio.h>
#include<conio.h>
#include<Windows.h>
#include<malloc.h>
#define EndofNumber 13
#define LastElement i+2

void rev_arr(int *arr)
{
int temp[12], i, max, j;
for (i = 0; arr[i] != EndofNumber; i++);
/*Counting the total number of elements except the last EndofNumber*/
max = i - 1;
/*Max is the total number of elements,
as the counting starts from 0 so we subtract 1 from it*/
i = 0;
for (j = max; j >= 0; j--, i++)
temp[j] = arr[i];/*temp starting from the last*/
for (i = 0; i <= max; i++)
arr[i] = temp[i];/*reversing the */
}
int main()
{
int num, a = 0, arr[12], i;
char word[90];
printf("\nEnter the number (max. 9 digits) : ");
scanf("%d", &num);
for (i = 0; num != 0; i++)
{
arr[i] = num % 10;
num /= 10;
}
}

```

```

/*Seperating the digits and saving the number in array*/
arr[i] = EndofNumber; /*Terminating array*/
rev_arr(arr); /*The numebrs in the array are saving in the reverse mode
as the saperation of the digits are takes place from last,
so we will reverse the numbers of the array but EndofNumber will have the
same
position*/
printf("\n\n\nNumber in words : ");
for (i = 0; arr[i] != EndofNumber; i++) /*Traversing the whole array.*/
{
    if(arr[i+5]==EndofNumber||arr[i+7]==EndofNumber||arr[i+
9]==EndofNumber||arr[i+2]==EndofNumber)
    {
        if (arr[i] == 1)
        {
            if (arr[i + 1] == 1)
                printf("Eleven ");
            if (arr[i + 1] == 2)
                printf("Twelve ");
            if (arr[i + 1] == 3)
                printf("Thirteen ");
            if (arr[i + 1] == 4)
                printf("Fourteen ");
            if (arr[i + 1] == 5)
                printf("Fifteen ");
            if (arr[i + 1] == 6)
                printf("Sixteen ");
            if (arr[i + 1] == 7)
                printf("Seventeen ");
            if (arr[i + 1] == 8)
                printf("Eighteen ");
            if (arr[i + 1] == 9)
                printf("Nineteen ");
            if (arr[i + 1] == 0)
                printf("Ten ");
        }
        i++;
    }
}
/*If we continue so these below words will not print as the loop is continueing
from here.*/
    if (arr[i + 10] == EndofNumber)
        printf("Arab ");
    if (arr[i + 8] == EndofNumber)
        printf("Crore ");
    if (arr[i + 6] == EndofNumber)
        printf("Lac ");
    if (arr[i + 4] == EndofNumber)
        printf("Thousand ");
    if (arr[i + 3] == EndofNumber)
        printf("Hundred ");
    continue;
}
if (arr[i] == 2)
    printf("Twenty ");
if (arr[i] == 3)
    printf("Thirty ");
if (arr[i] == 4)
    printf("Fourty ");
if (arr[i] == 5)
    printf("Fifty ");
if (arr[i] == 6)
    printf("Sixty ");
if (arr[i] == 7)

```

```

        printf("Seventy ");
    if (arr[i] == 8)
        printf("Eighty ");
    if (arr[i] == 9)
        printf("Ninety ");
}
else
{
    if (arr[i] == 1)
        printf("One ");
    if (arr[i] == 2)
        printf("Two ");
    if (arr[i] == 3)
        printf("Three ");
    if (arr[i] == 4)
        printf("Four ");
    if (arr[i] == 5)
        printf("Five ");
    if (arr[i] == 6)
        printf("Six ");
    if (arr[i] == 7)
        printf("Seven ");
    if (arr[i] == 8)
        printf("Eight ");
    if (arr[i] == 9)
        printf("Nine ");
}
if (arr[i + 10] == EndofNumber)
    printf("Arab ");
if (arr[i + 8] == EndofNumber)
    printf("Crore ");
if (arr[i + 6] == EndofNumber)
    printf("Lac ");
if (arr[i + 4] == EndofNumber)
    printf("Thousand ");
if (arr[i + 3] == EndofNumber)
    printf("Hundred ");
}
printf("\n");
_getch();
return 0;
}
(1)
#include<stdio.h>
#include<conio.h>
#include<Windows.h>

#define VisualStudio
//If using Visual Studio so delete its comment and make above two macro co
mments
//
#define TurboC //If using Turbo C so delete its comment and make above two mac
ro comments
//
#define DevC //If using Dev C so delete its comment and make above two macro c
omments
#define EndofNumber 13/*For Terminating the array just like strings do*/
int arr[10], x=0,one_i=0,two_i=0,three_i=0,four_i=0,five_i=0,six_i=
0,seven_i=0,eight_i=0,nine_i=0,zero_i=0;
/
*These variables are used in the multiple functions so I defined them globally
*/

```

```

/*These functions contains the pattern of # makes numbers*/
void one();
void two();
void three();
void four();
void five();
void six();
void seven();
void eight();
void nine();
void zero();
void jumpto();/
/*This functions calls the functions which is the next number*/
void rev_arr(int*);/*It will reverse the array*/
void rev_arr(int *array)
{
    int temp[12], i, max, j;
    for (i = 0; array[i] != EndofNumber; i++);
    /*Counting the total number of elemeents excpet the last EndofNumber*/
    max = i - 1;
    /*Max is the total number of elements,
    as the counting starts from 0 so we subtract 1 from it*/
    i = 0;
    for (j = max; j >= 0; j--, i++)
        temp[j] = array[i];/*temp starting from the last*/
    for (i = 0; i <= max; i++)
        array[i] = temp[i];/*reversing the */
}
void jumpto()
{
    if (arr[x] == 0)
        zero();
    else if (arr[x] == 1)
        one();
    else if (arr[x] == 2)
        two();
    else if (arr[x] == 3)
        three();
    else if (arr[x] == 4)
        four();
    else if (arr[x] == 5)
        five();
    else if (arr[x] == 6)
        six();
    else if (arr[x] == 7)
        seven();
    else if (arr[x] == 8)
        eight();
    else if (arr[x] == 9)
        nine();
    if (arr[x] == EndofNumber)
    {
        /*First it will print the first line of the each number,
        then by increamenting these in loops of individual number will
        print the next line*/
        one_i++;
        two_i++;
        three_i++;
        four_i++;
        five_i++;
        six_i++;
        seven_i++;
    }
}

```



```

eight_i++;
nine_i++;
zero_i++;
x = 0; /*When the first line of all the numbers has been printed then array
will initializes again so that we traverse the whole array again for pri
nting the */
printf("\n"); /*Jump to next line*/
jumpto();
}
return;
}
void one()
{
int j, k;
for (; one_i < 8; one_i++)
{
    if (one_i != 1 && one_i != 7)
        printf(" ");
    if (one_i == 1)
        k = 1;
    else if (one_i == 7)
        k = 2;
    else k = 0;
    for (j = 0; j <= k; j++)
        printf("#");
    x++;
    printf("\t");
    jumpto();
}
}
void two()
{
int j, k;
for (; two_i <= 7; two_i++)
{
    if (two_i > 0 && two_i < 4)
        printf(" ");
    if (two_i == 0 || two_i == 4 || two_i == 7)
        k = 4;
    else
        k = 0;
    for (j = 0; j <= k; j++)
        printf("#");
    x++;
    printf("\t");
    jumpto();
}
}
void three()
{
int j, k;
for (; three_i < 8; three_i++)
{
    if (three_i == 1 || three_i == 2 || three_i == 4 || three_i == 5 || th
ree_i == 6)
        printf(" ");
    if (three_i == 0 || three_i == 3 || three_i == 7)
        k = 4;
    else
        k = 0;
    for (j = 0; j <= k; j++)
        printf("#");
}
}

```

```

        x++;
        printf("\t");
        jumpto();
    }
}
void four()
{
    int j, k;
    for (; four_i < 8; four_i++)
    {
        if (four_i == 3)
            printf("# #");
        if (four_i == 5 || four_i == 6 || four_i == 7)
            printf(" ");
        if (four_i == 4)
            k = 4;
        else if (four_i == 3)
            k = -1;
        else
            k = 0;
        for (j = 0; j <= k; j++)
            printf("#");
        x++;
        printf("\t");
        jumpto();
    }
}
void five()
{
    int j, k;
    for (; five_i < 8; five_i++)
    {
        if (five_i > 3 && five_i < 7)
            printf(" ");
        if (five_i == 0 || five_i == 3 || five_i == 7)
            k = 4;
        else
            k = 0;
        for (j = 0; j <= k; j++)
            printf("#");
        x++;
        printf("\t");
        jumpto();
    }
}
void six()
{
    int j, k;
    for (; five_i < 8; five_i++)
    {
        if (five_i > 3 && five_i < 7)
            printf("# #");
        if (five_i == 0 || five_i == 3 || five_i == 7)
            k = 4;
        else if (five_i > 3 && five_i < 7)
            k = -1;
        else
            k = 0;
        for (j = 0; j <= k; j++)
            printf("#");
        x++;
        printf("\t");
    }
}

```

```

        jumpto();
    }
}
void seven()
{
for (; seven_i < 8; seven_i++)
{
    if (seven_i != 0)
        printf(" ");
    if (seven_i == 0)
        printf("#####");
    else
        printf("#");
    x++;
    printf("\t");
    jumpto();
}
}

void eight()
{
for (; eight_i < 8; eight_i++)
{
    if (eight_i == 0 || eight_i == 3 || eight_i == 7)
        printf("#####");
    else
        printf("#  #");
    x++;
    printf("\t");
    jumpto();
}
}

void nine()
{
int j, k;
for (; eight_i < 8; eight_i++)
{
    if (eight_i < 3 && eight_i > 0)
        printf("#  #");
    if (eight_i == 4 || eight_i == 5 || eight_i == 6)
        printf(" ");
    if (eight_i == 0 || eight_i == 7 || eight_i == 3)
        k = 4;
    else if (eight_i < 3 && eight_i > 0)
        k = -1;
    else
        k = 0;
    for (j = 0; j <= k; j++)
        printf("#");
    x++;
    printf("\t");
    jumpto();
}
}

void zero()
{
for (; nine_i < 8; nine_i++)
{
    if (nine_i == 0 || nine_i == 7)
        printf("#####");
    else
        printf("#  #");
    x++;

```

```

        printf("\t");
        jumpto();
    }
}
int main()
{
    int num, i;
    printf("\nEnter a five digit number : ");
    scanf("%d", &num);
    for (i = 0; num != 0; i++)
    {
        arr[i] = num % 10;
        num /= 10;
    }
    arr[i] = EndofNumber; /*Terminating the number*/
    rev_arr(arr);
    /
    *Reversing the array again as the number saved in it is already in reverse mode*/

```

```

#ifdef TurboC
clrscr();
#endif
#ifdef VisualStudio
system("cls");
#endif
#ifdef DevC
system("cls");
#endif
printf("\n\n\n");
jumpto();
_getch();
return 0;
}

```

<https://github.com/hiverkiya/Let-Us-C-Solutions>

Structures

14 July 2020 16:14

[Chapter 17: Structures] Solutions

[A]

(a) 1004 1008 1058

(b)

```
CrankShaft 102133
CrankShaft 102133
CrankShaft 102133
```

(c) 2 If you are driven by success make sure that it is a quality drive

[B]

(a) age cannot be used without any object reference.

(b) book is not any structure name.
b do not belong to any structure.

(c) variable v in printf statement, should have some index,
as it is an array of structure.

(d)

1. In f() function, structure is passed by value, so "." operator should be used.

2. In g() function, structure is passed by its address, so "->" operator should be used.

(e) NO ERROR

[C]

(a) Array

(b) 2 is true.

(c) True,False,True.

(d) 2 & 4

(e)

1. = g. 25

2. = h. 8
i[7] = 2, s[7] = x=8/y=75,
so s[7].x = 8

3. = b. 2
3[i] = i[3] = 15,
(s+2)->y = 75,
75/15 = 3,
s[3].y = 2

4. = a. 75

```
i[1] = 20, i[2] = 15,
20-15 = 5,
i[5] = 75
```

```
5. = f. 15
s[3] = 2
i[2] = 15
```

```
6. = l. 20
```

```
7. = e. 16
*(1+i) = 20,
*(i+4) = 8,
*i = 10,
20*8/10 = 16
```

```
8. = a. 85
i[0] = 10, i[4] = 8
s[10-8].y = 75,
75+10 = 85
```

```
9. = k. 10
*(i+1) = 20, *i = 10,
s+(20/2) = x=8/y=75,
*(s+2).x = 8,
8+2 = 10
```

```
10. = d. 7
i[6] = 6,
++6 = 7
```

[D]

(a)

```
#include<stdio.h>
#include<conio.h>
#include<Windows.h>
struct studata
{
int roll;
char name[50];
char depart[50];
char course[20];
int yoj;
}nos[450] = { 03, "Kishor", "IT", "B.tech", 2014,
10, "Mohammad Siraj Alam", "Computer Science", "B.Tech", 2014,
53, "Deepak", "Computer Applications", "BCA", 2013,
13, "Karan", "Mechanical", "Diploma", 2013,
1, "Aakash", "Tool and Die", "Diploma", 2012,
2, "Ramesh", "IT", "B.tech", 2014
};
void student_in_year(int year)
{
int i;
printf("\n\t\tYear of joining : %d\n\n", year);
for (i = 0; i <= 450; i++)
{
if (nos[i].yoj == year)
{
printf("\nRoll Number : %d", nos[i].roll);
printf("\nName : %s", nos[i].name);
printf("\nDepartment : %s", nos[i].depart);
```

```

        printf("\nCourse : %s\n", nos[i].course);
    }
}
}
void student_data(int enroll)
{
    int i;
    printf("\nRoll number : %d", enroll);
    for (i = 0; i <= 450; i++)
    {
        if (nos[i].roll == enroll)
        {
            printf("\nName : %s", nos[i].name);
            printf("\nDepartment : %s", nos[i].depart);
            printf("\nCourse : %s\n", nos[i].course);
            printf("\nYear of joining : %d\n", nos[i].yoj);
        }
    }
}
}
int main()
{
    int yoj, roll;
    printf("\nEnter year of joining of the students : ");
    scanf("%d", &yoj);
    student_in_year(yoj);
    _getch();
    system("cls");
    printf("\nEnter the roll number of the student you want data : ");
    scanf("%d", &roll);
    student_data(roll);
    _getch();
    return 0;
}

```

(b)

```

#include<stdio.h>
#include<conio.h>
#include<Windows.h>
/*Function to perform withdrawal or deposition*/
void action(int, int, int);
/*Print the balance below 100 Rs.*/
void below100();
struct acc_holder
{
    long int acc_num;
    char name[30];
    int bal;
} sbi[200] = { 1, "Siraj", 1000000,
2, "Azad", 1233044,
3, "Deepak", 99,
4, "Rihan", 33,
5, "Rahul Khawal", 200000
};
int main()
{
    int accnum, amount, code;
    printf("\nEnter your account number : ");
    scanf("%d", &accnum);
    printf("Enter 1 for deposit and 0 for withdrawal : ");
    scanf("%d", &code);
    if (code)
    {
        printf("\nEnter amount to be deposit : ");
    }
}

```

```

        scanf("%d", &amount);
    }
    else
    {
        printf("\nEnter amount to withdraw : ");
        scanf("%d", &amount);
    }
    action(accnum, amount, code);
    _getch();
    system("cls");
    printf("All members with account balance less than 100 are following : ");
    below100();
    _getch();
    return 0;
}
void below100()
{
    int i;
    for (i = 0; i < 200; i++)
    {
        if (sbi[i].bal < 100 && sbi[i].bal > 0)
        {
            printf("\nName : %s", sbi[i].name);
            printf("\nAccount Number : %d\n\n", sbi[i].acc_num);
        }
    }
}
void action(int accnum, int amount, int code)
{
    int i;
    for (i = 0; i < 200; i++)
        if (sbi[i].acc_num == accnum)
            break;
    if (!code)
    {
        if (sbi[i].bal - amount < 100)
        {
            printf("\nThe balance is insufficient for the specified withdrawal
");
            return;
        }
        else
        {
            sbi[i].bal -= amount;
            printf("\nYour new account balance is : %d", sbi[i].bal);
        }
    }
    else
    {
        sbi[i].bal += amount;
        printf("\nYour new account balance is : %d", sbi[i].bal);
    }
}
(c)
#include<stdio.h>
#include<conio.h>
#include<Windows.h>
void eng_info(char*, char*);
struct engine
{
    char serial[4];
    int yom;

```



```

char mat[50];
int quantity;
}
maruti[10] = { "AA0", 2005, "Iron", 20,
"BB1", 2007, "Steel", 13,
"BB2", 1992, "Aluminium", 57,
"CC1", 2005, "Stainless Steel", 7,
"CC6", 2007, "Steel", 34,
"CC7", 2010, "Steel", 14
};
int main()
{
char from[5], to[5];
printf("\nEnter the serial number, from where you want to start the list :
");
scanf("%s", from);
printf("\nEnter the serial at which you wan to end the list : ");
scanf("%s", to);
system("cls");
eng_info(from, to);
_getch();
return 0;
}
void eng_info(char *from, char *to)
{
char first_letter;
int last_digit;
int i;
printf("\n\t\tEngines Information\n");
for (i = 0; maruti[i].serial[0] != *from; i++);
while (1)
{
printf("\nSerial Number : %s", maruti[i].serial);
printf("\nYear Of Manufacture : %d", maruti[i].yom);
printf("\nMaterail Used : %s", maruti[i].mat);
printf("\nQuantity : %d\n", maruti[i].quantity);
if (i == 10)
i = 0;
i++;
if (maruti[i].serial[0] == *to && maruti[i].serial[2] == *(to + 2))
{
printf("\nSerial Number : %s", maruti[i].serial);
printf("\nYear Of Manufacture : %d", maruti[i].yom);
printf("\nMaterail Used : %s", maruti[i].mat);
printf("\nQuantity : %d\n", maruti[i].quantity);
return;
}
}
}
}
(d)
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
#define Max 20
int compare(const void * a, const void * b);
struct cricketers
{
int avrun;
char name[30];
int age;
int notm;
}india[Max] = {

```

```

122, "Sachin Tendulkar", 30, 67,
97, "Virendra Sehwag", 35, 56,
66, "Irfan Pathan", 32, 45,
153, "Yusuf Pathan", 36, 21,
101, "Yuvaraj Singh", 32, 45,
};
int main()
{
int i;
qsort(india, 5, sizeof(struct cricketers), compare);
/*qsort() is a standard library function.

```

view the link below for more info

http://www.tutorialspoint.com/c_standard_library/c_function_qsort.htm

```

*/
for (i = 0; i < 5; i++)
{
    printf("Name : %s", india[i].name);
    printf("\nAge : %d", india[i].age);
    printf("\nTotal Test Matches played : %d", india[i].notm);
    printf("\nAverage Run : %d\n\n", india[i].avrun);
}
_getch();
return 0;
}
int compare(const void * a, const void * b)
{
return (*(int*)a - *(int*)b);
}

```

(e)

```

#include<stdio.h>
#include<conio.h>
void printinfo(struct employee k);
struct employee
{
int code;
char name[30];
int doj[3];
}hcl[50] = {
001, "Shahnawaz", 13, 1,2006,
004, "Amit Puri", 21, 6,2008,
102, "Irfan Moin", 12, 5,2012,
131, "Shabnam", 16, 1, 2014
};
int main()
{
int i, d[3];
printf("\nEnter the current date (dd mm yyyy) : ");
scanf("%d%d%d", &d[0], &d[1], &d[2]);
printf("\nEmployees with greater than or equal to 3 years of tenure\n\n");
for (i = 0; i < 4; i++)
{
    if (d[2] - hcl[i].doj[2] > 3)
        printinfo(hcl[i]);
    else if (d[2] - hcl[i].doj[2] == 3)
    {
        if (d[1] - hcl[i].doj[1] > 0)
            printinfo(hcl[i]);
        else if (hcl[i].doj[1] == d[1])
        {

```

```

        if (d[0] - hcl[i].doj[0] >= 0)
            printinfo(hcl[i]);
    }
}
}
_getch();
return 0;
}
void printinfo(struct employee e)
{
    printf("\nCode : %d", e.code);
    printf("\nName : %s", e.name);
    printf("\nDate of joining : %d-%d-%d\n", e.doj[0], e.doj[1], e.doj[2]);
}
(f)
#include<stdio.h>
#include<conio.h>
#include<string.h>
#include<Windows.h>
/*Count total number of books in the library*/
int count();
void display(int);
/*If the author name is same so the function strcmp returns a zero*/
void giv_author(char*);
void adbuk();
/*Appears the title of the specified book*/
void which1(int);
/*Displays all the books serially accession number*/
void albuk();
struct library
{
    int an; /*Accession number*/
    char title[50];
    char author[30];
    int price;
    int flag; /*flag = 0 : Issued and, flag = 1: not issued*/
}geclib[50] = {
    1, "Let Us C", "Yashavant Kanethkar", 258, 1,
    2, "Data Structure Through C", "Yashavant Kanethkar", 300, 1,
    3, "Let Us C++", "Yashavant Kanethkar", 220, 1,
    4, "Harry Potter : The Philosopher's Stone", "J. K. Rowling", 550, 0,
    5, "The Two Towers", "J. R. R. Tolkien", 560, 0,
    6, "The Hobbit", "J. R. R. Tolkien", 550, 1,
    7, "The Fellowship of the Ring", "J. R. R. Tolkien", 550, 0
};
int main()
{
    int ans = 1, an;
    char auth[30];
    while (ans != 7)
    {
        system("cls");
        printf("\nYou are in menu section\n");
        printf("\n1. Add Book Information");
        printf("\n2. Display Book Information");
        printf("\n3. List all books of given author");
        printf("\n4. List the title of specified book");
        printf("\n5. List the counts of the books in library");
        printf("\n6. List the books in order of accession number");
        printf("\n7. Exit");
        printf("\n\n\t\tEnter you choice : ");
        scanf("%d", &ans);
    }
}

```

```

switch (ans)
{
case 1:
    adbuk();
    _getch();
    break;
case 2:
    printf("\nEnter the accession number of the book : ");
    scanf("%d", &an);
    display(an);
    _getch();
    break;
case 3:
    while (getchar() != '\n');
    printf("Enter the name of the author (case sensitive) : ");
    gets_s(auth);
    giv_author(auth);
    _getch();
    break;
case 4:
    printf("\nEnter the accession number of book : ");
    scanf("%d", &an);
    which1(an);
    _getch();
    break;
case 5:
    printf("\nTotal Number of books : %d", count());
    _getch();
    break;
case 6:
    albuk();
    _getch();
    break;
case 7:
    return 0;
default:
    printf("\nWrong choice, Try Again!!");
    _getch();
}
}
_getch();
return 0;
}
int count()
{
    int i = 0;
    while (geclib[i].an)
        i++;
    return i;
}
void display(int i)
{
    i--; /*To use i as index*/
    printf("\n\n\n");
    printf("\nAccession Number : %d", geclib[i].an);
    printf("\nTitle : %s", geclib[i].title);
    printf("\nAuthor : %s", geclib[i].author);
    printf("\nPrice : %d", geclib[i].price);
    if (geclib[i].flag) /*Flag : 0*/
        printf("\nStatus : Issued");
    else
        printf("\nStatus : Available"); /*Flag : 1*/
}

```

```

}

void giv_author(char *author)
{
int i = 0;
printf("\nBooks of \"%s\" are following : \n\n", author);
while (geclib[i].an)
{
    if (!(strcmp(author, geclib[i].author)))
        display(geclib[i].an);
    i++;
}
}

void adbuk()
{
int next = count();
geclib[next].an = next + 1;
/*fflush(stdin) or while(getch() != '\n'); to clear the buffer*/
while (getchar() != '\n');
printf("\nEnter the title of the book : ");
gets_s(geclib[next].title);
printf("\nEnter the author name of the book : ");
gets_s(geclib[next].author);
printf("\nEnter the price of the book : ");
scanf("%d", &geclib[next].price);
geclib[next].flag = 1;
system("cls");
}

void which1(int an)
{
int i = 0;
while (geclib[i].an)
{
    if (geclib[i].an == an)
    {
        printf("\n\nTitle of the book : %s\n", geclib[i].title);
        return;
    }
}
printf("No any book found found\n");
}

void albuk()
{
int i = 0;
while (geclib[i].an)
{
    display(i + 1);
    i++;
}
}

(g)
#include<stdio.h>
#include<conio.h>
struct dmy
{
int date;
int month;
int year;
};
int datcmp(struct dmy a, struct dmy b)
{
if (a.date == b.date && a.month == b.month && a.year == b.year)

```

```

        return 0;
    else
        return 1;
    }
    int main()
    {
        struct dmy a, b;
        int flag;
        printf("\nEnter the first date (dd mm yyyy) : ");
        scanf("%d%d%d", &a.date, &a.month, &a.year);
        printf("\nEnter the second date (dd mm yyyy) : ");
        scanf("%d%d%d", &b.date, &b.month, &b.year);
        flag = datcmp(a, b);
        if (flag)
            printf("\nThe dates are not same\n");
        else
            printf("\nThe dates are same\n");
        _getch();
        return 0;
    }
}
(h)
#include<stdio.h>
#include<conio.h>
#include<malloc.h>
#include<stdlib.h>
#include<Windows.h>
#define GetSize (struct node*)malloc(sizeof(struct node))
void insert(struct node**, int);
void del(struct node**, int);
void display(struct node*);
struct node
{
    int data;
    struct node* next;
};
int main()
{
    char ans = 'y';
    int num, exit = 1, action;
    struct node *start = NULL;
    printf("\nHello all, this program is of linked list, so go ahead and make
your linked list\n\n");
    while (ans == 'y')
    {
        printf("\nEnter the data : ");
        scanf("%d", &num);
        insert(&start, num);
        printf("\nWant to enter another number (y/n) : ");
        while (getchar() != '\n');
        scanf("%c", &ans);
    }
    system("cls");
    while (exit != 0)
    {
        system("cls");
        display(start);
        printf("\n\n\nYou are in menu section\n\n");
        printf("\n0 : Exit\n1 : Insert\n2 : Delete\n3 : Display");
        printf("\n\n\nEnter your choice : ");
        scanf("%d", &action);
        switch (action)
        {

```

```

    case 0:
        printf("\nYou are exiting. . . .");
        _getch();
        return 0;
    case 1:
        printf("\nEnter number : ");
        scanf("%d", &num);
        insert(&start, num);
        break;
    case 2:
        printf("\nEnter number to delete : ");
        scanf("%d", &num);
        del(&start, num);
        break;
    case 3:
        display(start);
        break;
    default:
        printf("\nYou have entered a wrong option\,Try againA!!\n");
        exit = 1;
}
}
_getch();
return 0;
}

void insert(struct node **start, int num)
{
    struct node *help = NULL, *curr = NULL;
    help = GetSize;
    help->data = num; /*This is our new node.*/
    help->next = NULL;

    if (*start == NULL) /*Inserting the first element*/
    {
        *start = help;
        return;
    }
    /*Finding the position for the number to insert*/
    for (curr = *start; curr->next != NULL; curr = curr->next);
    curr->next = help;
}

void del(struct node **start, int num)
{
    struct node *curr = *start, *pre = NULL;
    /*Finding the data to be deleted */
    while (curr->data != num)
    {
        pre = curr;
        curr = curr->next;
        /*List ends and the number nor found.*/
        if (curr == NULL)
            break;
    }
    /*Deleting the first node*/
    if (curr == *start)
        *start = curr->next;
    else if (curr == NULL) /*Number not found so just return*/
    {
        printf("\nNumber is not present in the list.");
        return;
    }
    else /*Deleting in between or last node*/

```

```

        pre->next = curr->next;
        free(curr);/*Deletion*/
    }
    void display(struct node *start)
    {
        struct node *i = NULL;
        printf("\nYour list\n");
        for (i = start; i != NULL; i = i->next)
            printf("%d\t", i->data);
    }
(i)
#include<stdio.h>
#include<conio.h>
#include<malloc.h>
#include<stdlib.h>
#include<Windows.h>
#define GetSize (struct node*)malloc(sizeof(struct node))
void push(struct node**, int);
int pop(struct node**);
struct node
{
    int data;
    struct node* next;
};
int main()
{
    char ans = 'y';
    int num, exit = 1, action;
    struct node *top = NULL;
    printf("\nHello all, this program is of stacks using linked list, so");
    printf("go ahead and make your linked list that will saved in stacks.\n\n");
    while (ans == 'y')
    {
        printf("\nEnter the data : ");
        scanf("%d", &num);
        push(&top, num);
        printf("\nWant to enter another number (y/n) : ");
        while (getchar() != '\n');
        scanf("%c", &ans);
    }
    system("cls");
    while (exit != 0)
    {
        printf("\n\n\nYou are in menu section\n\n");
        printf("\n0 : Exit\n1 : Push\n2 : Pop");
        printf("\n\n\nEnter your choice : "); +
            scanf("%d", &action);
        switch (action)
        {
            case 0:
                printf("\nYou are exiting. . . .");
                _getch();
                return 0;
            case 1:
                printf("\nEnter number : ");
                scanf("%f", &num);
                push(&top, num);
                system("cls");
                break;
            case 2:
                system("cls");

```



```

        if (num == NULL)
        {
            printf("\nStack is empty");
            break;
        }
        num = pop(&top);
        printf("\n\n%d is popped.", num);
        break;
    default:
        printf("\nYou have entered a wrong option\,Try againA!!\n");
        exit = 1;
    }
}
_getch();
return 0;
}
/*Push means adding an element in the stack*/
void push(struct node **top, int num)
{
    struct node *temp = NULL, *help = NULL;
    help = GetSize;
    if (help == NULL)
    {
        printf("\nStack is full!!");
        return;
    }
    help->data = num;
    help->next = *top;
    /*Last added element is always the top element.*/
    *top = help;
}
/*Pop means isolating the top element from the stacks.*/
int pop(struct node **top)
{
    struct node *temp = GetSize;
    int num;
    if (*top == NULL)
        return NULL;
    num = (*top)->data;
    temp = *top;
    *top = (*top)->next;
    free(temp);
    return num;
}
(j)
#include<stdio.h>
#include<conio.h>
#include<malloc.h>
#include<stdlib.h>
#include<Windows.h>
#define GetSize (struct node*)malloc(sizeof(struct node))
void add(struct node**, struct node**, int);
int del(struct node**, struct node**);
void display(struct node*);
struct node
{
    int data;
    struct node* next;
};
int main()
{
    char ans = 'y';

```

```

int num, exit = 1, action;
struct node *start = NULL, *end = NULL;
printf("\nHello all, this program is of queue using linked list, ");
printf("so go ahead and make your linked list that wil saved in a queue.\n
\n");
while (ans == 'y')
{
    printf("\nEnter the data : ");
    scanf("%d", &num);
    add(&start, &end, num);
    printf("\nWant to enter another number (y/n) : ");
    while (getchar() != '\n');
    scanf("%c", &ans);
}
display(start);
system("cls");
while (exit != 0)
{
    display(start);
    printf("\n\nYou are in menu section\n\n");
    printf("\n0 : Exit\n1 : Add\n2 : Remove\n3 : Display");
    printf("\n\nEnter your choice : "); +
        scanf("%d", &action);
    switch (action)
    {
    case 0:
        printf("\nYou are exiting. . .");
        _getch();
        return 0;
    case 1:
        printf("\nEnter number : ");
        scanf("%d", &num);
        add(&start, &end, num);
        display(start);
        system("cls");
        break;
    case 2:
        system("cls");
        if (num == NULL)
        {
            printf("\nQueue is empty!!");
            break;
        }
        num = del(&start, &end);
        printf("\n%d has been extracted.\n\n", num);
        break;
    case 3:
        display(start);
        break;
    default:
        printf("\nYou have entered a wrong option\,Try againA!!\n");
        exit = 1;
    }
}
_getch();
return 0;
}

void add(struct node **start, struct node **end, int num)
{
    struct node *help = NULL;
    help = GetSize;
    help->data = num;

```

```

help->next = NULL;
if (*start == NULL)
{
    *start = help;
    *end = help;
    return;
}
(*end)->next = help;
*end = help;
}

int del(struct node **start, struct node **end)
{
    struct node *temp;
    int num;
    temp = GetSize;
    if (*start == NULL)
        return NULL;
    temp = *start;
    num = temp->data;
    *start = (*start)->next;
    free(temp);
    return num;
}
void display(struct node *start)
{
    struct node *a = start;
    while (a != NULL)
    {
        printf("%d\t", a->data);
        a = a->next;
    }
}

```

(k)

```

#include<stdio.h>
#include<conio.h>
#include<malloc.h>
#include<stdlib.h>
#include<Windows.h>
#define GetSize (struct node*)malloc(sizeof(struct node))
void insert(struct node**, int);
void del(struct node**, int);
void display(struct node*);
struct node
{
    int data;
    struct node* next;
};
int main()
{
    char ans = 'y';
    int num, exit = 1, action;
    struct node *start = NULL;
    printf("\nHello all, this program is of linked list, so go ahead and make
your linked list\n\n");
    while (ans == 'y')
    {
        printf("\nEnter the data : ");
        scanf("%d", &num);
        insert(&start, num);
        printf("\nWant to enter another number (y/n) : ");
    }
}

```

```

        while (getchar() != '\n');
        scanf("%c", &ans);
    }
    system("cls");
    while (exit != 0)
    {
        display(start);
        printf("\n\nYou are in menu section\n\n");
        printf("\n0 : Exit\n1 : Insert\n2 : Delete\n3 : Display");
        printf("\n\nEnter your choice : ");
        scanf("%d", &action);
        switch (action)
        {
            case 0:
                printf("\nYou are exiting. . . .");
                _getch();
                return 0;
            case 1:
                system("cls");
                printf("\nEnter number : ");
                scanf("%d", &num);
                insert(&start, num);
                display(start);
                break;
            case 2:
                system("cls");
                printf("\nEnter number to delete : ");
                scanf("%d", &num);
                del(&start, num);
                display(start);
                break;
            case 3:
                display(start);
                break;
            default:
                printf("\nYou have entered a wrong option\,Try againA!!\n");
                exit = 1;
        }
    }
    _getch();
    return 0;
}

void insert(struct node **start, int num)
{
    struct node *help = NULL, *curr = NULL, *pre = NULL;
    help = GetSize;
    help->data = num; /*This is our new node.*/
    if (*start == NULL) /*Inserting the first element*/
    {
        help->next = NULL;
        *start = help;
        return;
    }
    for (curr = *start; curr->data <= num; pre = curr, curr = curr->next)
        if (curr->next == NULL)
            break;

    if (curr == *start && num < curr->data)
    {
        help->next = *start;
        *start = help;
    }
}

```

```

else if (curr->next == NULL && num > curr->data)
{
    curr->next = help;
    help->next = NULL;
}
else/*Insertion in between*/
{
    pre->next = help;
    help->next = curr;
}
}

void del(struct node **start, int num)
{
    struct node *curr = *start, *pre = NULL;
    while (curr->data != num)
    {
        pre = curr;
        curr = curr->next;
        if (curr == NULL)
            break;
    }
    if (curr == *start)
        *start = curr->next;
    else if (curr == NULL)
    {
        printf("\nNumbe is not present in the list.");
        return;
    }
    else
        pre->next = curr->next;
    free(curr);
}

void display(struct node *start)
{
    struct node *i = NULL;
    printf("\nYour list\n");
    for (i = start; i != NULL; i = i->next)
        printf("%d\t", i->data);
}
}

(1)
#include<stdio.h>
#include<conio.h>
int main()
{
    int wind;
    char cat;
    printf("\nEnter the speed of the wind (in miles/hour) : ");
    scanf("%d", &wind);
    if (wind < 73)
        printf("\nCool down man, this isn't a hurricane");
    else if (wind > 73 || wind < 96)
        cat = 'I';
    else if (wind > 95 || wind < 111)
        cat = 'II';
    else if (wind > 110 || wind < 131)
        cat = 'III';
    else if (wind > 130 || wind < 156)
        cat = 'IV';
    else
        cat = 'V';
    printf("\nThe wind speed belongs to the Hurricane Category '%c'", cat);
}

```

```

    _getch();
    return 0;
}
(m)
#include<stdio.h>
#include<conio.h>
#include<windows.h>
struct marks
{
    int j[3];
    int tot;
};
    int main()
{
    struct marks player[5];
    int i, j, max = 0;
    for (i = 0; i < 5; i++)
    {
        system("cls");
        printf("\n\t\tMarks for player-%d", i + 1);
        for (j = 0; j < 3; j++)
        {
            printf("\nEnter marks entered by judege number %d : ", j + 1);
            scanf("%d", &player[i].j[j]);
        }
        player[i].tot = player[i].j[0] + player[i].j[1] + player[i].j[2];
        if (player[i].tot > max)
            max = player[i].tot;
    }
    for (i = 0; i < 5; i++)
    {
        if (max == player[i].tot)
        {
            printf("\nPlayer number %d is choosen as MVP\n", i + 1);
            _getch();
            return 0;
        }
    }
}

```

<https://github.com/hiverkiya/Let-Us-C-Solutions>

Console I/P O/P

14 July 2020 16:14

[Console Input/Output] Solutions

[A]

(a)

If upper case letter is entered so lower case of it will be printed otherwise if lower case is entered so uppercase will be printed.

(b)

2 2.537 Life is like that

(c)

the person who wins is the one who thinks he can!

(d)

The sixth sick sheikh's sixth ship is sick

[B]

(a)

'a' is an array of characters, which is initialized at the time of declaration, so it cannot be modified. i.e. writing 'a++' is illegal.

(b)

Wrong format specifier used, in scanf()

(c)

n is undefined in scanf().

(d)

1. i is undefined.
2. We cannot save string in a char type pointer from scanf() function.

(e) No error

(f)

sprintf() first argument should be of array of characters char*, instead of char type.

(g)

No error but the warning, "format specifier" should not be used in scanf() or sscanf() functions.

[C]

(a) 2. gets(str);

(b) 2. getche()

(c) 1. scanf()

(d)1. `getch()` : This is a function defined in file `conio.h` and used to get a single character from keyboard, without displaying it on screen, and no need to press enter after you entered the character.
 2. `getche()` : This is a function same as `getch()`, but the only is that `getche()` displays the character you enter on screen when you hit the button while `getch()` do the same without displaying it on screen. `e` in `getche()` function means echoes means displays. It's prototype is present in `conio.h`

3. `getchar()` : `getchar()` is not a function, but it is an macro, present in `stdio.h` file. It is used to get character from keyboard after pressing enter.

4. `fgetchar()` : `fgetchar()` is same as `getchar()`, the only difference between them is that `fgetchar()` is a function while `getchar()` is the macro, both are present in the same file `stdio.h`

(e)1. Characters, format specifications and escape sequences
 Because : format specifications manage the space that a number or float will take and escape sequences move the cursor and do many formatting in an array of characters i.e. strings.

(f)4. Specifies how many columns will be used to print the number.

[D]

(a)

```
#include<stdio.h>
void xgets(char *str)
{
    scanf("%[^\n]s", str);
}
void xputs(char *str)
{
    printf("%s\n", str);
}
```

(b)

```
#include<stdio.h>
#include<conio.h>
int getint();
int main()
{
    int num;
    num = getint();
    printf("\nNumber : %d\n", num);
    _getch();
    return 0;
}
int getint()
{
    int num = 0;
    char number[20];
    scanf("%s", number);
    sscanf(number, "%d", &num);
    return num;
}
```

(c)


```

#include<stdio.h>
#include<conio.h>
double getfloat();
int main() /*Main is written to check the getfloat function*/
{
double num;
num = getfloat();
printf("\nNumber : %lf\n", num);
_getch();
return 0;
}
double getfloat()
{
double num = 0;
char number[20];
scanf("%s", number);
sscanf(number, "%lf", &num);
return num;
}

```

(d)

```

For first column : %26s
For second column : %18s
For third column : %3.2lf

```

<https://github.com/hiverkiya/Let-Us-C-Solutions>

File IO

14 July 2020 16:14

[Chapter 19: File Input/Output] Solutions

[A]

(a)

Argument of `openfile()` is of `FILE**` type, while in main `FILE*` is passed in the function.

(b)

1. `NULL` is not defined, `NULL` should be used at that place.
2. `exit()` function should have an `int` type argument, which is not present there.
3. `putch()` is defined in `conio.h` file which is also not present.

(c)

"tr" is not any valid file opening mode, it should be "r", "rt" or "rb". And a .dat should be open in binary mode, that is in "rb" mode.

(d)

1. `fgets()` do not return EOF, so while loop statement is invalid.
2. We are storing 81 characters in `str` variable, 80 characters form the file and 1 character is added automatically the `NULL` character that caused overflow of the array.
3. `fputs()` accepts three arguments, `char*` and `FILE*`, but only one argument is passed.

(e)

`ch` is not defined in this scope.

(f)

1. `sizeof` operator do not have spce between it.
2. `close()` function is undefined. Use `fclose()`;

(g)

1. Cannot use `fwrite()` function on the file opened in "r" (read only) mode.
2. `close()` function is undfined. Use `fclose()`;

(h)

1. `open()` and `close()` functions are UNIX platofrm specific. Use `_open()` or `_close()` and include `io.h` file if you are using Visual Studio;

(i)

1. `close()` and `open()` functions are not defined, with the same reason stated above.
2. `fopen()` returns `FILE*` but it is assigned to an `int` type variable.
3. `READ` | `BINARY` are undefined

[B]

(a)

4. `stdio.h`

(b)

3. I am a boy\n\0

(c)

T,T,T,F.

(d)

4. All the above.

(e)

No, it is not necessary that a file is created in text mode must always be opened in text mode, it can also be opened in binary mode, for subsequent operations. Binary or text depends upon the content storage mode, a file with extension .txt can also be a binary file.

(f)

1. If "myfile.c" exists in the disk, so simple FILE pointer fp points to the file.
2. If "myfile.c" does not exist in the disk, so as the opening mode is "r" read only, so no new file is created and fopen() returns a NULL assigned to fp.

(g)

When we get any character from keyboard, so this procedure takes place.

- We enter character from keyboard.
- Characters entered are going into Standard Input Stream
- function like scanf() or getch() fetch that character from the Standard Input Stream, and not from the keyboard.

So, hitting Enter key after the character key, store Enter also in Standard Input Stream, so the next function scanf() or getch(), do not wait to get character from the keyboard, but take the Enter key, hit previously from the Standard Input Stream. So, to solve this problem we have to clear our buffer completely before functions scanf("%

c") or getch().

This function is performed by fflush(stdin).

(h)

1. If 'myfile.c' does not exist in the disk, so a new file is created, and fp points to that new file in binary mode.
2. If 'myfile.c' exists on the disk, so fp points to that file in binary mode.

(i)

If we have to store something in a file, so we should open this file in write mode. And opening the file in binary mode, saves lots of space and works faster, so I will choose the mode "wb+".

[C]

(a)

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
int main()
{
    FILE *fp;
    int i = 1;
    char ch;
    fp = fopen("File (a).txt", "r+");
```

```

if (fp == NULL)
{
    puts("\nFile can't be open");
    exit(1);
}
/*Line no. before first line*/
printf("%d. ", i);
i++;
do
{
    ch = fgetc(fp);
    if (ch == '.')
    {
        printf("%c", ch);
        if (fgetc(fp) == EOF)
            continue;
        /*Printing the line number*/
        printf("\n%d. ", i);
        i++;
        continue;
    }
    printf("%c", ch);
} while (ch != EOF);
fclose(fp);
_getch();
return 0;
}

```

(b)

```

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
int main()
{
    FILE *f1, *f2;
    char ch, buffer[10];
    //Destination File
    f1 = fopen("File (b)1.txt", "a+");
    if (f1 == NULL)
    {
        printf("File do not exist or can't be open");
        exit(1);
    }
    //Source File
    f2 = fopen("File (b)2.txt", "r+");
    if (f2 == NULL)
    {
        printf("File do not exist or can't be open");
        exit(2);
    }
    fseek(f1, 0, SEEK_END);
    //Goes to last character
    ch = fgetc(f2);
    while (ch != EOF)
    {
        fputc(ch, f1);
        ch = fgetc(f2);
    }
    rewind(f1);
    while (fgets(buffer, 9, f1) != NULL)/*Displaying the content*/
        printf("%s", buffer);
    fclose(f1);
}

```

```

fclose(f2);
_getch();
return 0;
}

```

(c)

```

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
#include<string.h>
void sort_names(char**, int);
void swap(char*, char*);
int main()
{
char ch = 'i', *help[30], name[50], *p;
int i, j, tot_names;
FILE *fp;
fp = fopen("File (c).txt", "r+");
if (fp == NULL)
{
printf("\nCannot open the file.\n");
exit(1);
}
for (i = 0; ch != EOF; i++)
{
ch = fgetc(fp);
for (j = 0; ch != '\n'; j++)
{
if (ch == EOF)
break;
name[j] = ch;
ch = fgetc(fp);
}
//To terminate the string
name[j] = '\0';
/*Saving the name in array of pointers*/
p = (char*)malloc(50);
strcpy(p, name);
help[i] = p;
}
tot_names = i - 1;
sort_names(help, tot_names);
//Displaying the content
for (j = 0; j <= tot_names; j++)
puts(help[j]);
fclose(fp);
_getch();
return 0;
}
void swap(char *a, char *b)
{
char temp[20];
strcpy(temp, a);
strcpy(a, b);
strcpy(b, temp);
}
void sort_names(char** name_list, int tot_names)
{
int i, j, k = 0;
for (i = 0; i <= tot_names; i++)
{
for (j = i + 1; j <= tot_names; j++)

```

```

    {
        /*If the name starts with same letter,
        so it checks the next letter of that name*/
        for (k = 0; name_list[i][k] == name_list[j][k]; k++);
        if (name_list[i][k] > name_list[j][k])
            swap(name_list[j], name_list[i]);
    }
}
}
(d)

```

```

#include<stdio.h>
#include<conio.h>
#include<string.h>
int main()
{
    char str[50];
    FILE *f1, *f2;
    f1 = fopen("File (d)1.txt", "r");
    f2 = fopen("File (d)2.txt", "w");
    while (fgets(str, 49, f1) != NULL)
    {
        _strupr(str);
        fputs(str, f2);
    }
    _getch();
    return 0;
}
(e)

```

```

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
int main()
{
    FILE *f1, *f2, *fp;
    char ch1 = 'a', ch2 = 'a';
    f1 = fopen("File (e)1.txt", "r");
    f2 = fopen("File (e)2.txt", "r");
    fp = fopen("File (e)3.txt", "w");
    if (f1 == NULL)
    {
        printf("Can't open the file1\n");
        exit(1);
    }
    if (f2 == NULL)
    {
        printf("Can't open the file1\n");
        exit(2);
    }
    puts("\nWork on progress\n.\n.\n.\n.\n");
    while (1)
    {
        if (ch1 != EOF)
        {
            ch1 = fgetc(f1);
            /*A line is ends when a . is encounter*/
            while (ch1 != '.')
            {
                if (ch1 == EOF)
                    break;
            }
        }
    }
}

```

```

        fputc(ch1, fp);
        ch1 = fgetc(f1);
    }
    if (ch1 != EOF)
        fputc('.', fp);
}
if (ch2 != EOF)
{
    ch2 = fgetc(f2);
    /*A line is ends when a . is encounter*/
    while (ch2 != '.')
    {
        if (ch2 == EOF)
            break;
        fputc(ch2, fp);
        ch2 = fgetc(f2);
    }
    if (ch2 != EOF)
        fputc('.', fp);
}
/*Getting out of the loop after end of both files*/
if (ch1 == EOF && ch2 == EOF)
    break;
}
printf("\nTask completed.\nExiting . . . \n");
_getch();
return 0;
}

```

(f)

```

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
#include<Windows.h>
void getkey();
void gotoxy(short, short);
void box(short, short, short, short);
int main()
{
    box(0, 1, 79, 23);
    FILE *fp;
    char str[85];
    int i, j = 1;
    fp = fopen("File (f).txt", "r");
    if (fp == NULL)
    {
        printf("File do not exist, or can't open the file.");
        exit(1);
    }
    i = 2;
    while (fgets(str, 76, fp) != NULL)
    {
        gotoxy(1, 0);
        /*Printing page number*/
        printf("%d", j);
        gotoxy(32, 0);
        /*Printing the file name.*/
        printf("File (f).txt");
        gotoxy(3, i);
        /*Printing the file's content.*/
        printf("%s", str);
        i++;
    }
}

```

```

    if (i == 20)
    {
        gotoxy(32, i + 4);
        printf("Press any key...");
        getch();
        system("cls");
        box(0, 1, 79, 23);
        /*Re-initializing variable for new page*/
        i = 2;
        /*Page number increamentation*/
        j++;
    }
}
getch();
system("cls");
gotoxy(3, 10);
printf("File Ends\n");
_getch();
return 0;
}
void getch()
{
    char ch;
    ch = _getch();
    if (ch == 0)
        ch = _getch();
}
void gotoxy(short col, short row)
{
    HANDLE h = GetStdHandle(STD_OUTPUT_HANDLE);
    COORD position = { col,row };
    SetConsoleCursorPosition(h, position);
}
void box(short x1, short y1, short x2, short y2)
{
    int i;
    gotoxy(0, 1);
    printf("%c", 218);/*Making upper-left corener*/
    gotoxy(79, 1);
    printf("%c", 191);/*Making upper-right corner*/
    gotoxy(0, 23);
    printf("%c", 192);/*Making lower-left corner*/
    gotoxy(79, 23);
    printf("%c", 217);/*Making lower-right corner*/

    for (i = x1 + 1; i < x2; i++)
    {
        gotoxy(i, y1);
        printf("%c", 196);/*This will make upper side of the box*/
        gotoxy(i, y2);
        printf("%c", 196); /*This will make lower side of the box*/
    }
    for (i = y1 + 1; i < y2; i++)
    {
        gotoxy(x1, i);
        printf("%c", 179); //This will make left side of the box
        gotoxy(x2, i);
        printf("%c", 179); //This will make right side of the box
    }
}
}

```

(g)


```

#include<stdio.h>
#include<conio.h>
#include<ctype.h>
void encode(FILE*, FILE*);
void decode(FILE*, FILE*);
void display(FILE*);
int main()
{
FILE *sf, *df;
/*Source File*/
sf = fopen("File (g)1.txt", "r");
/*Destination File*/
df = fopen("File (g)2.txt", "w");
/*Displaying decoded file before encoding*/
display(sf);

/*Encoding file*/
encode(sf, df);
/*Displaying encoded file after encoding*/
display(df);

/*Decoding file*/
decode(df, sf);
/*Displaying file after decoding*/
display(sf);
_getch();
return 0;
}
void encode(FILE *sf, FILE *df)
{
char ch;
while (1)
{
    ch = fgetc(sf);
    if (ch == EOF)
        break;
    if (isalpha(ch) || ch == ' ' || ch == '\n')
    {
        switch (ch)
        {
            case '\n':
                fputc('@', df);
                break;
            case 'a':
                fputc('0', df);
            case 'A':
                fputc('!', df);
                break;
            case 'b':
                fputc('0', df);
            case 'B':
                fputc('#', df);
                break;
            case 'c':
                fputc('0', df);
            case 'C':
                fputc('$', df);
                break;
            case 'd':
                fputc('0', df);
            case 'D':
                fputc('%', df);

```

```

        break;
case 'e':
    fputc('0', df);
case 'E':
    fputc('&', df);
    break;
case 'f':
    fputc('0', df);
case 'F':
    fputc('*', df);
    break;
case 'g':
    fputc('0', df);
case 'G':
    fputc('(', df);
    break;
case 'h':
    fputc('0', df);
case 'H':
    fputc(')', df);
    break;
case 'i':
    fputc('0', df);
case 'I':
    fputc('+', df);
    break;
case 'j':
    fputc('0', df);
case 'J':
    fputc('/', df);
    break;
case 'k':
    fputc('0', df);
case 'K':
    fputc('{', df);
    break;
case 'l':
    fputc('0', df);
case 'L':
    fputc('}', df);
    break;
case 'm':
    fputc('0', df);
case 'M':
    fputc(';', df);
    break;
case 'n':
    fputc('0', df);
case 'N':
    fputc(':', df);
    break;
case 'o':
    fputc('0', df);
case 'O':
    fputc('[', df);
    break;
case 'p':
    fputc('0', df);
case 'P':
    fputc(']', df);
    break;
case 'q':

```

```

        fputc('0', df);
case 'Q':
    fputc('"', df);
    break;
case 'r':
    fputc('0', df);
case 'R':
    fputc('`', df);
    break;
case 's':
    fputc('0', df);
case 'S':
    fputc('\\', df);
    break;
case 't':
    fputc('0', df);
case 'T':
    fputc('|', df);
    break;
case 'u':
    fputc('0', df);
case 'U':
    fputc('=', df);
    break;
case 'v':
    fputc('0', df);
case 'V':
    fputc('-', df);
    break;
case 'w':
    fputc('0', df);
case 'W':
    fputc('_', df);
case 'x':
    fputc('0', df);
case 'X':
    fputc('<', df);
    break;
case 'y':
    fputc('0', df);
case 'Y':
    fputc('^', df);
    break;
case 'z':
    fputc('0', df);
case 'Z':
    fputc('?', df);
    break;
case ' ':
    fputc('1', df);
    break;
    }
}
else
    fputc(ch, df);
}
}

void decode(FILE *ef, FILE *df)
{
    char ch;
    while (1)
    {

```

```

ch = fgetc(df);
if (ch == EOF)
    break;
switch (ch)
{
case '@':
    fputc('\n', ef);
    break;
case '0':
    ch = fgetc(df);
    switch (ch)
    {
    case '!':
        fputc('a', ef);
        break;
    case '#':
        fputc('b', ef);
        break;
    case '$':
        fputc('c', ef);
        break;
    case '%':
        fputc('d', ef);
        break;
    case '&':
        fputc('e', ef);
        break;
    case '*':
        fputc('f', ef);
        break;
    case '(':
        fputc('g', ef);
        break;
    case ')':
        fputc('h', ef);
        break;
    case '+':
        fputc('i', ef);
        break;
    case '/':
        fputc('j', ef);
        break;
    case '{':
        fputc('k', ef);
        break;
    case '}':
        fputc('l', ef);
        break;
    case ';':
        fputc('m', ef);
        break;
    case ':':
        fputc('n', ef);
        break;
    case '[':
        fputc('o', ef);
        break;
    case ']':
        fputc('p', ef);
        break;
    case '"':
        fputc('q', ef);

```

```

        break;
    case '\':
        fputc('r', ef);
        break;
    case '\\':
        fputc('s', ef);
        break;
    case '|':
        fputc('t', ef);
        break;
    case '=':
        fputc('u', ef);
        break;
    case '-':
        fputc('v', ef);
        break;
    case '_':
        fputc('w', ef);
        break;
    case '<':
        fputc('x', ef);
        break;
    case '^':
        fputc('y', ef);
        break;
    case '?':
        fputc('z', ef);
        break;
    }
    break;
case '!':
    fputc('A', ef);
    break;
case '#':
    fputc('B', ef);
    break;
case '$':
    fputc('C', ef);
    break;
case '%':
    fputc('D', ef);
    break;
case '&':
    fputc('E', ef);
    break;
case '*':
    fputc('F', ef);
    break;
case '(':
    fputc('G', ef);
    break;
case ')':
    fputc('H', ef);
    break;
case '+':
    putc('I', ef);
    break;
case '/':
    fputc('J', ef);
    break;
case '{':
    fputc('K', ef);

```

```

        break;
    case '}':
        fputc('L', ef);
        break;
    case ';':
        fputc('M', ef);
        break;
    case ':':
        fputc('N', ef);
        break;
    case '[':
        fputc('O', ef);
        break;
    case ']':
        fputc('P', ef);
        break;
    case '"':
        fputc('Q', ef);
        break;
    case '\\':
        fputc('R', ef);
        break;
    case '\\':
        fputc('S', ef);
        break;
    case '|':
        fputc('T', ef);
        break;
    case '=':
        fputc('U', ef);
        break;
    case '-':
        fputc('V', ef);
        break;
    case '_':
        fputc('W', ef);
    case '<':
        fputc('X', ef);
        break;
    case '^':
        fputc('Y', ef);
        break;
    case '?':
        fputc('Z', ef);
        break;
    case '1':
        fputc(' ', ef);
        break;
    default:
        fputc(ch, ef);
}
}
}
void display(FILE *fp)
{
    char str[50];
    while (fgets(str, 49, fp) != NULL)
        printf("%s", str);
    puts("\n\n");
    _getch();
}

```

(h)

```

#include<stdio.h>
#include<Windows.h>
#include<string.h>
#include<conio.h>
void display(char*);
void add_info(int, char*, float);
void transaction(int, char, float);
struct customer
{
int accno;
char name[30];
float balance;
};
struct trans
{
int accno;
char trans_type;
float amount;
};
int main()
{
add_info(1, "Siraj", 1000);
puts("\n\t\tBefore Transaction");
display("customer.dat");
transaction(1, 'd', 1000);
puts("\n\t\tAfter Transaction");
display("customer.dat");
_getch();
return 0;
}
void display(char *file)
{
FILE *fp;
struct customer holder;
fp = fopen(file, "rb");
while (fread(&holder, sizeof(holder), 1, fp) == 1)
{
printf("\n%d", holder.accno);
printf(": \t%s", holder.name);
printf("\t%f\n", holder.balance);
}
fclose(fp);
}
void add_info(int accno, char *name, float bal)
{
FILE *fp;
struct customer holder;
fp = fopen("customer.dat", "rb+");
if (fp == NULL)
fp = fopen("customer.dat", "wb");
fseek(fp, 0, SEEK_END);
holder.accno = accno;
strcpy(holder.name, name);
holder.balance = bal;
fwrite(&holder, sizeof(holder), 1, fp);
fclose(fp);
}
void transaction(int accno, char ttype, float amount)
{
FILE *fp, *temp;
struct customer holder;

```

```

fp = fopen("customer.dat", "rb");
temp = fopen("temp.dat", "wb");
while (fread(&holder, sizeof(holder), 1, fp) == 1)
{
    if (holder.accno == accno)
    {
        switch (ttype)
        {
            case 'd':
            case 'D':
                holder.balance += amount;
                break;
            case 'w':
            case 'W':
                if ((holder.balance - amount) < 100)
                {
                    system("cls");
                    printf("\nYour account balance is low.\n");
                    printf("Transaction failed!!");
                    _getch();
                }
                else
                {
                    holder.balance -= amount;
                    break;
                }
            default:
                system("cls");
                puts("Wrong transaction type!!");
                puts("\nTry Again!!");
                fclose(fp);
                fclose(temp);
                remove("temp.dat");
                return;
            }
        }
        fwrite(&holder, sizeof(holder), 1, temp);
    }
    fclose(fp);
    fclose(temp);
    remove("customer.dat");
    rename("temp.dat", "customer.dat");
}

```

(i)

```

#include<stdio.h>
#include<conio.h>
#include<string.h>
#include<Windows.h>
struct date
{
    int d, m, y;
};
struct employee
{
    char empcode[6];
    char empname[20];
    struct date join_date;
    float salary;
};
/*To display list in way they are actually saved in disk.*/
void display(char*);
/*To make file before sorting them*/
void add_info(char*, char*, struct date, float);

```



```

/*Sort by date of joining*/
void sortbydoj(char*);
void swap(struct employee*, struct employee*);
int main()
{
    struct date d = {13, 1, 1995};
    add_info("EMP01", "Siraj", d, 50000.00);
    d.d = 17; d.m = 5; d.y = 1994;
    add_info("EMP04", "Karan", d, 45000.00);
    d.d = 7; d.m = 7; d.y = 1995;
    add_info("EMP03", "Deepak", d, 45000.00);
    d.d = 2; d.m = 11; d.y = 1995;
    add_info("EMP02", "Azad", d, 45000.00);
    d.d = 15; d.m = 1; d.y = 2001;
    add_info("EMP04", "Vijay", d, 21000.00);
    d.d = 24; d.m = 5; d.y = 1993;
    add_info("EMP06", "Shanu", d, 55000.00);
    sortbydoj("Record.dat");
    _getch();
    return 0;
}

void display(char *file)
{
    FILE *fp;
    struct employee e;
    fp = fopen(file, "rb");
    while (fread(&e, sizeof(e), 1, fp) == 1)
    {
        printf("\n%s", e.empcode);
        printf("\t%s", e.empname);
        printf("\t%d", e.join_date.d);
        printf("\t%d", e.join_date.m);
        printf("\t%d", e.join_date.y);
        printf("\t%f\n", e.salary);
    }
    fclose(fp);
}

void add_info(char *code, char *name, struct date doj, float salary)
{
    FILE *fp;
    fp = fopen("Record.dat", "rb+");
    if (fp == NULL)
        fp = fopen("Record.dat", "wb");
    struct employee e;
    strcpy(e.empcode, code);
    strcpy(e.empname, name);
    e.join_date = doj;
    e.salary = salary;
    fseek(fp, 0, SEEK_END);
    fwrite(&e, sizeof(e), 1, fp);
    fclose(fp);
}

void swap(struct employee *a, struct employee *b)
{
    struct employee temp;
    temp = *a;
    *a = *b;
    *b = temp;
}

void sortbydoj(char *file)
{

```

```

int i = 0, j, count;
FILE *fp1;
struct employee e[100], temp;
fp1 = fopen(file, "rb");
if (fp1 == NULL)
{
    puts("\nUnable to open file or the file do not exist.");
    exit(1);
}
while (1)
{
    /*Saving the data in the array of structures*/
    if (fread(&e[i], sizeof(e[i]), 1, fp1) != 1)
        break;
    i++;
}
/*Total number of records*/
count = i;
for (i = 0; i < count; i++)
{
    for (j = i + 1; j < count; j++)
    {
        if (e[i].join_date.y >= e[j].join_date.y)
            if (e[i].join_date.y > e[j].join_date.y)
                swap(&e[i], &e[j]);
            else
                if (e[i].join_date.m >= e[j].join_date.m)
                    if (e[i].join_date.m > e[j].join_date.m)
                        swap(&e[i], &e[j]);
                    else if (e[i].join_date.d >= e[j].join_date.d)
                        swap(&e[i], &e[j]);
    }
}

/*Printing the array after sorting by date of joining*/
printf("Employee Id\tName\tdd mm yyyy\tSalary\n");
for (i = 0; i < count; i++)
{
    printf("\n%-12s", e[i].empcode);
    printf("\t%s", e[i].empname);
    printf("\t % 2d", e[i].join_date.d);
    printf("% 2d", e[i].join_date.m);
    printf("% 2d", e[i].join_date.y);
    printf("\t%2.2f\n", e[i].salary);
}
}
(j)

```

```

#include<stdio.h>
#include<conio.h>
#include<windows.h>
#include<stdlib.h>
void add_info(struct blood);
void display(char*);
void lessthen25(char*);
struct blood
{
    char name[20];
    char address[40];
    int age;
    int blood_type;
};

```

```

int main()
{
    lessthen25("Blood Donors Record.txt");
    _getch();
    return 0;
}

void add_info(struct blood data)
{
    FILE *fp;
    fp = fopen("Blood Donors Record.txt", "r+");
    if (fp == NULL)
        fp = fopen("Blood Donors Record.txt", "w");
    fseek(fp, 0, SEEK_END);
    fprintf(fp, "%-20s", data.name);
    fprintf(fp, "%-40s ", data.address);
    fprintf(fp, "%-2d ", data.age);
    fprintf(fp, "%d\n", data.blood_type);
    fclose(fp);
}

void display(char *file)
{
    FILE *fp;
    fp = fopen(file, "r");
    char str[20];
    while (fgets(str, 19, fp) != NULL)
        printf("%s", str);
    fclose(fp);
}

void lessthen25(char *file)
{
    FILE *fp;
    struct blood data;
    int age;
    char str[68];
    fp = fopen(file, "r");
    while (fgets(str, 67, fp) != NULL)
    {
        age = (str[62] - 48) * 10 + (str[63] - 48);
        if (age < 25)
            printf("%s", str);
    }
    fclose(fp);
}

```

(k)

```

#include<stdio.h>
#include<conio.h>
#include<string.h>
#include<stdlib.h>
#include<Windows.h>
void add_info(char*, char*);
void display(char*);
void disbys(int, char*);
int main()
{
    disbys(1, "Students.txt");
    _getch();
    return 0;
}

void add_info(char *name, char *file)
{
    FILE *fp;

```

```

fp = fopen(file, "r+");
if (fp == NULL)
    fp = fopen(file, "w");
fseek(fp, 0, SEEK_END);
fputs(name, fp);
putc('\n', fp);
fclose(fp);
}
void display(char *file)
{
FILE *fp;
char name[30];
fp = fopen(file, "r");
if (fp == NULL)
{
    printf("File do not exist.");
    _getch();
    exit(1);
}
while (fscanf(fp, "%s", name) != -1)
    puts(name);
fclose(fp);
}
void disbys(int num, char *file)
{
FILE *fp;
int i;
char name[30];
fp = fopen(file, "r");
if (fp == NULL)
{
    printf("File do not exist.");
    _getch();
    exit(2);
}
for (i = 1; fscanf(fp, "%s", name) != -1; i++)
{
    if (i == num)
    {
        switch (num)
        {
            case 1:
                printf("%dst name is : ", num);
                break;
            case 2:
                printf("%dnd case is : ", num);
                break;
            case 3:
                printf("%drd name is : ", num);
                break;
            default:
                printf("%dth name is : ", num);
        }
        printf("%s\n\n", name);
        break;
    }
}
}
if (fscanf(fp, "%s", name) == -1)
    printf("There's only %d names.\n", i);
rewind(fp);
while (fscanf(fp, "%s", name) != -1)
    if (name[0] == 'S' || name[0] == 's')

```

```

        puts(name);
    fclose(fp);
}
(1)

#include<stdio.h>
#include<conio.h>
#include<string.h>
#include<Windows.h>
#include<stdlib.h>

struct data
{
    int rollno;
    char name[20];
};
void add_info(int, char*);
void display(char*);
void update_list(char*);
int main()
{
    add_info(1, "Azad Ansari");
    add_info(2, "Deepak Mathpal");
    add_info(3, "Rahul Khawal");
    add_info(4, "Siraj");
    add_info(5, "Priya Saxena");
    add_info(6, "Rajkumari");
    add_info(7, "Varun Taneja");
    add_info(8, "Manish Kumar");
    add_info(9, "Shabnam");
    update_list("Transaction.txt");
    display("New List.dat");
    _getch();
    return 0;
}
void add_info(int rollno, char *name)
{
    struct data e;
    FILE *fp;
    fp = fopen("Students.dat", "rb+");
    if (fp == NULL)
        fp = fopen("Students.dat", "wb");
    e.rollno = rollno;
    strcpy(e.name, name);
    fseek(fp, 0, SEEK_END);
    fwrite(&e, sizeof(e), 1, fp);
    fclose(fp);
}
void display(char *fname)/* Display the MASTER FILE*/
{
    FILE *fp;
    struct data e;
    fp = fopen(fname, "rb");
    if (fp == NULL)
    {
        puts("Can't open the file.\nExiting...\n");
        _getch();
        exit(3);
    }
    while (fread(&e, sizeof(e), 1, fp) == 1)
        printf("%2d\t%-10s\n", e.rollno, e.name);
}

```

```

void update_list(char *tfile)
{
    FILE *tf, *mf, *uf;
    char code, name[20];
    int rollno, i;
    struct data dat_of_mf;
    tf = fopen(tfile, "r");
    if (tf == NULL)
    {
        printf("\nTransaction file is not found");
        printf("\nexiting...\n");
        _getch();
        exit(1);
    }
    mf = fopen("Students.dat", "rb");
    if (mf == NULL)
    {
        printf("\nMaster file is not found\nexiting...\n");
        _getch();
        exit(2);
    }
    uf = fopen("New List.dat", "wb+");
    while (1)
    {
        code = fgetc(tf);
        if (code == 'D')
        {
            fseek(tf, 1, SEEK_CUR);
            rollno = fgetc(tf) - 48;
            while (1)
            {
                fread(&dat_of_mf, sizeof(dat_of_mf), 1, mf);
                if (dat_of_mf.rollno < rollno)
                    fwrite(&dat_of_mf, sizeof(dat_of_mf), 1, uf);
                else
                    break;
            }
            while (fgetc(tf) == '\n');
            fseek(tf, -1, SEEK_CUR);
        }
        else if (code == 'A')
        {
            /* Skipping the space after trasnsaction code*/
            fseek(tf, 1, SEEK_CUR);
            rollno = fgetc(tf) - 48;
            /*Skipping the space after trasnsaction code*/
            fseek(tf, 1, SEEK_CUR);
            for (i = 0; 1; i++)
            {
                name[i] = fgetc(tf);
                if (name[i] == '\n' || name[i] == EOF)
                {
                    name[i] = '\0';
                    break;
                }
            }
            dat_of_mf.rollno = rollno;
            strcpy(dat_of_mf.name, name);
            fwrite(&dat_of_mf, sizeof(dat_of_mf), 1, uf);
        }
        else
        {

```

```

        while (fread(&dat_of_mf, sizeof(dat_of_mf), 1, mf) == 1)
            fwrite(&dat_of_mf, sizeof(dat_of_mf), 1, uf);
        break;
    }
}
fclose(uf);
fclose(mf);
fclose(tf);
}
(m)

```

```

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
#include<string.h>
#include<Windows.h>
struct empo
{
int eid;
char name[20];
char sex;
int salary;
};
int main()
{
addemp(1, "Kishor Dass", 'M', 100);
addemp(2, "Amit Kumar Mandal", 'M', 200);
addemp(3, "Prakash Babu Sharma", 'M', 300);
addemp(4, "Mohd Seraj Alam", 'M', 400);
addemp(5, "Priya Saxena", 'F', 500);
addemp(6, "Aabha Chaudhary", 'F', 600);
addemp(7, "Varun Tanjea", 'M', 700);
addemp(8, "Ankit Bhardwaj", 'M', 800);
change_sal(1, 1000);
disemp();
_getch();
return 0;
}
void addemp(int eid, char *name, char sex, int salary)
{
struct empo help;
FILE *fp;
fp = fopen("Emp.txt", "rb+");
if (fp == NULL)
    fp = fopen("Emp.txt", "wb");
fseek(fp, 0, SEEK_END);
help.eid = eid;
strcpy(help.name, name);
help.sex = sex;
help.salary = salary;
fwrite(&help, sizeof(help), 1, fp);
fclose(fp);
}
void change_sal(int id, int new_salary)
{
struct empo help;
int size = sizeof(help);
FILE *fp;
fp = fopen("Emp.txt", "rb+");
if (fp == NULL)
    fp = fopen("Emp.txt", "wb");
while (fread(&help, sizeof(help), 1, fp) == 1)

```

```

    {
        if (help.eid == id)
        {
            fseek(fp, -size, SEEK_CUR);
            help.salary = new_salary;
            fwrite(&help, sizeof(help), 1, fp);
            break;
        }
    }
    fclose(fp);
}
void delemp(int id)
{
    struct empo help;
    int size = sizeof(help);
    FILE *fp;
    fp = fopen("Emp.txt", "rb+");
    if (fp == NULL)
        fp = fopen("Emp.txt", "wb");
    while (fread(&help, sizeof(help), 1, fp) == 1)
    {
        if (help.eid == id)
        {
            fseek(fp, -size, SEEK_CUR);
            help.salary = 0;
            fwrite(&help, sizeof(help), 1, fp);
            break;
        }
    }
    fclose(fp);
}
void disemp()
{
    struct empo help;
    FILE *fp;
    fp = fopen("Emp.txt", "rb");
    if (fp == NULL)
    {
        puts("\nFile is unable to open or did not exist.\nExiting. . .");
        exit(1);
    }
    while (fread(&help, sizeof(help), 1, fp) == 1)
    {
        printf("\n%2d", help.eid);
        printf("\t%-20s", help.name);
        printf("\t%c", help.sex);
        printf("\t%d\n", help.salary);
    }
    fclose(fp);
}
(n)

```

```

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
#include<string.h>
#include<Windows.h>
#define FOUND (strcmp(word, "a") == 0 || strcmp(word, "an") == 0\
|| strcmp(word, "the") == 0 || strcmp(word, "A") == 0\
|| strcmp(word, "An") == 0 || strcmp(word, "The") == 0)
void delart(char*);
int main()

```



```

{
delart("Temp.txt");
_getch();
return 0;
}
void delart(char *file)
{
FILE *fp, *nf;
int i;
char word[20];
fp = fopen(file, "r+");
nf = fopen("NEW FILE.txt", "w");
if (fp == NULL)
{
puts("\nCannot open the file.\n");
exit(1);
}
for (i = 0; 1; i++)
{
word[i] = fgetc(fp);
if (word[i] == EOF)
break;
if (word[i] == ' ')
{
word[i] = '\\0';
if (FOUND)
{
/*Replacing the word with the space*/
strcpy(word, " ");
fputs(word, nf);
}
else
{
fputs(word, nf);
fputc(' ', nf);
}
i = -1;
/*After the loop ends, i increament by 1,
so -1 becomes 0 and next character saved
in the word[0]*/
}
}
}
}
(o)

```

```

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
#include<string.h>
#include<Windows.h>
struct employee
{
int empno;
char name[30];
int basic, grade;
};
void addemp(int, char*, int, int, char*);
void display(FILE*);
void miss(FILE*);
int main()
{
FILE *fp;

```

```

addemp(1,"Kishor Dass", 100, 1, "File (o).txt");
addemp(2,"Amit Kumar Mandal", 101,2, "File (o).txt");
addemp(3,"Prakash babu Sharma", 100, 3, "File (o).txt");
addemp(4, "Mohd Seraj Alam", 123, 1, "File (o).txt");
addemp(5, "Priya Saxena", 100, 2, "File (o).txt");
addemp(9, "Varun Taneja", 120, 3, "File (o).txt");
addemp(13, "Shahrukh", 123, 3, "File (o).txt");
addemp(20, "Paras Chugh", 452, 1, "File (o).txt");
puts("\nList of the employees.\n\n");
fp = fopen("File (o).txt", "rb+");
display(fp);
_getch();
system("cls");
miss(fp);
_getch();
fclose(fp);
return 0;
}

void addemp(int eid, char *name, int bas, int grad, char *file)
{
FILE *fp;
struct employee dat;
fp = fopen(file, "rb+");
if (fp == NULL)
{
    fp = fopen(file, "wb");
    if(fp == NULL)
    {
        printf("\nCannot open the file.\n");
        exit(1);
    }
}
dat.empno = eid;
strcpy(dat.name, name);
dat.basic = bas;
dat.grade = grad;
fseek(fp, 0, SEEK_END);
fwrite(&dat, sizeof(dat), 1, fp);
fclose(fp);
}

void display(FILE *fp)
{
struct employee dat;
while (fread(&dat, sizeof(dat), 1, fp) == 1)
{
    printf("%2d", dat.empno);
    printf("\t%-20s", dat.name);
    printf("\t % 4d", dat.basic);
    printf("\t % 3d\n", dat.grade);
}
}

void miss(FILE *fp)
{
int i = 1;
struct employee dat;
rewind(fp);
puts("\nEmployees with following IDs are missing.");
while (fread(&dat, sizeof(dat), 1, fp) == 1)
{
    while(dat.empno != i)
    {
        printf("%d, ", i);
    }
}
}

```

```

        i++;
    }
    i++;
}
}
(p)

```

```

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
#include<string.h>
#include<Windows.h>
#define EndOfWord word[i] == ' ' || word[i] == ','\
    || word[i] == '.' || word[i] == '\n'
int count_word(FILE*);
void count_4_words(FILE*);
int main()
{
    FILE *fp;
    fp = fopen("File (p).txt", "r");
    if (fp == NULL)
    {
        puts("\nCan't open the file.");
        exit(1);
    }
    int count = count_word(fp);
    printf("\nThere are total %d words in file.\n", count);
    _getch();
    system("cls");
    count_4_words(fp);
    _getch();
    return 0;
}
int count_word(FILE *fp)
{
    int i, count = 0;
    char word[20];
    for (i = 0; 1; i++)
    {
        word[i] = fgetc(fp);
        if (word[i] == EOF)
            break;
        /*A word ends if a space or comma or etc. encounter.*/
        if (EndOfWord)
        {
            /*If the 'word' variable contain only one letter and viz.
            '.' or ',' or ' ' or '\n' so it skips it as
            it is not count as a word*/
            if (i != 0)
                count++;
            i = -1;
        }
    }
    return count;
}
fclose(fp);
}
void count_4_words(FILE *fp)
{
    int i, count = 0;
    char word[20];
    if (fp == NULL)
    {

```

```

        puts("\nCan't open the file.");
        exit(1);
    }
    puts("\nTotal number of four letter words are following.\n");
    for (i = 0; 1; i++)
    {
        word[i] = fgetc(fp);
        if (word[i] == EOF)
            break;
        /*A word ends if a space or comma or etc. encounter.*/
        if (EndOfWord)
        {
            /*If the 'word' variable contain only one letter and viz.
            '.' or ',' or ' ' or '\n' so it skips it as
            it is not count as a word*/
            if (i != 0)
            {
                if (i == 4)
                {
                    /*Terminating the word before printing it*/
                    word[i + 1] = '\0';
                    puts(word);
                }
            }
            i = -1;
        }
    }
    fclose(fp);
}

```

(q)

```

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
#include<string.h>
#include<Windows.h>
#define EndOfWord word[i][j] == ' ' || word[i][j] == ',' \
|| word[i][j] == '.' || word[i][j] == '\n'
void swap(char*, char*);
void list(FILE *);
int main()
{
    FILE *fp;
    fp = fopen("File (q).txt", "r");
    if (fp == NULL)
    {
        puts("\nCannot open the file.\n");
        exit(1);
    }
    list(fp);
    _getch();
    return 0;
}
void swap(char *one, char *two)
{
    char three[20];
    strcpy(three, one);
    strcpy(one, two);
    strcpy(two, three);
}
void list(FILE *fp)
{
    char word[30][20];

```

```

int i = 0, j, count = 0, k1 = 0, k2 = 0;
/* 'count' is used to count total number of words in the list,
and k1 and k2 are used to locate k1th and k2th characters
in Ist and IInd words respectively and 'i' is the first word
and 'j' is second*/
for (j = 0; 1; j++)
{
    word[i][j] = fgetc(fp);
    if (word[i][j] == EOF)
        break;
    /*A word may be ends with a space or a comma etc.*/
    if (EndOfWord)
    {
        if (j != 0)
        {
            word[i][j] = '\0';
            i++;
            /*It means, first word is successfully saved
            in word[i][j]. Increament in i takes 2-d array
            in next 1d array and save next word in it.*/
            count++;
        }
        /*If a word saved, so second word should be saved
        with second index 0.*/
        j = -1;
    }
}

/* Loop below is for sorting the words by alphabatical order a to z,
and all words should have only upper or only lower case words.*/
for (i = 0; i < count; i++)
{
    for (j = i + 1; j < count; j++)
    {
        if (word[i][k1] > word[j][k2])
        {
            swap(word[i], word[j]);
            k1 = k2 = 0;
            /*If second word is smaller alphabatically,
            so it will come in first by swapping them together*/
            continue;
        }
        if (word[i][k1] < word[j][k2])
        {
            k1 = k2 = 0;
            /*If first word is smaller alphabatically, so next word
            is compared without making any changes*/
            continue;
        }
        if (word[i][k1] == word[j][k2])
        {
            /*If both the words are identical, so no action is taken*/
            if (word[i][k1 + 1] == '\0' && word[j][k2 + 1] == '\0')
            {
                k1 = k2 = 0;
                continue;
            }
            /* If one word is end and second is not so next character is
            checked of second word with the first character of first word,
            that means, only k2 increases and k1 remains same */
            if (word[i][k1 + 1] != '\0')
                k1++;
        }
    }
}

```

```

        if (word[j][k2 + 1] != '\0')
            k2++;
        /*Next character is checked of the same word, so j should be
        decreament as it is also increamenting after every loop*/
        j--;
    }
}
}
system("cls");
printf("\nTotal number of words in the list is : %d\n", count);
puts("Alphabetical listing of words is : ");
for (i = 0; i < count; i++)
    printf("\n%-2d : %s", i + 1, word[i]);
fclose(fp);
}
(r)

```

```

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
#include<string.h>
void frev(FILE*);
int main()
{
    FILE *fp;
    fp = fopen("File (r).txt", "r");
    if (fp == NULL)
    {
        puts("File cannot be opened.\n");
        exit(1);
    }
    frev(fp);
    fclose(fp);
    _getch();
    return 0;
}
void frev(FILE *fp)
{
    char word[11];
    int i;
    for (i = 0; 1; i++)
    {
        word[i] = fgetc(fp);
        if (word[i] == EOF)
            break;
        if (word[i] == ' ' || word[i] == '\n')
        {
            /*Convert that space/newline to NULL character
            to complete the word for printing*/
            word[i] = '\0';

            _strrev(word);
            printf("%s ", word);
            /*When i increament so i becomes 0 and a
            new word start to save in the word variable*/
            i = -1;
        }
    }
}
(s)

```

```

#include<stdio.h>

```

```

#include<conio.h>
#include<stdlib.h>
#include<string.h>
#include<Windows.h>
void pagebreak(FILE *fp);
int main()
{
FILE *fp;
fp = fopen("NOTE.txt", "r");
if (fp == NULL)
{
    puts("Can't open the file.");
    exit(1);
}
pagebreak(fp);
fclose(fp);
_getch();
return 0;
}

void pagebreak(FILE *fp)
{
/*Dot counts the numebr of lines*/
int dot = 0;
char ch;
while (1)
{
    ch = fgetc(fp);
    if (ch == EOF)
        break;
    printf("%c", ch);
    /*If a full stop appears so the line is terminated here*/
    if (ch == '.')
    {
        /*If dot == 50 so, 50 lines have been printed,
        so a page breakmsg should be appear here*/
        if (dot == 50)
        {
            printf("Press any key. . .");
            _getch();
            system("cls");
            /*Dot variable reinitialization*/
            dot = 1;
        }
        else
            dot++;
    }
}
}
}

```

<https://github.com/hiverkiya/Let-Us-C-Solutions>

More issues on IO

14 July 2020 16:15

[Chapter 20: More Issues on Input/Output] Solutions

[A]

(a)

Let the file name is `prog.c` so its executable file will be `prog.exe`

. Now,

type this command in the command prompt

(1) - To copy the content of one file into another.

`prog.exe > File1.txt < File2.txt`

Input from file1 and put output in file 2

(2)

To create a new file and add some text to it.

`prog.exe < NewFile.txt`

Input from keyboard and output in the file specified.

(3) To display contents of an existing file.

`prog.exe > File1.txt`

Input from the file1 and output in the screen

(b)

False,True,True

(c)

Syntax error in second argument

[B]

(a)

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
#include<stdlib.h>
#define EndOfWord word[i] == ' ' || word[i] == ',' \
|| word[i] == '\n' || word[i] == '.'
int main(int ac, char *av[])
{
    FILE *fp, *temp;
    int i, j;
    char word[10], ch;
    if (ac != 4)
    {
        puts("Wrong number of arguments\n.");
        printf("Write like this : ");
        printf("\nchange \"old_word\"");
        printf("\nnew_word \"file_name\"");
        exit(1);
    }
    fp = fopen(av[3], "r");
    temp = fopen("temp.txt", "w");
    if (fp == NULL)
    {
        perror("Error : ");
        exit(1);
    }
```



```

for (i = 0; (word[i] = fgetc(fp)) != EOF; i++)
{
    if (EndOfWord)
    {
        ch = word[i];
        word[i] = '\0';
        if (!strcmp(av[1], word))
            fputs(av[2], temp);
        else
            fputs(word, temp);
        fputc(ch, temp);
        i = -1;
    }
}
fclose(fp);
fclose(temp);
remove(av[3]);
rename("temp.txt", av[3]);
return 0;
}

```

(b)

```

#include<stdio.h>
#include<conio.h>
#include<string.h>
#include<stdlib.h>
int main(int ac, char *av[])
{
    int i, a, b;
    a = b = 0;
    if (ac != 4)
    {
        puts("Wrong number of arguments");
        printf("\n.Write like this : ");
        printf("\ncalc\ "switch\");
        printf("\number 1\ \number 2\");
        exit(1);
    }
    for (i = 0; av[2][i] != '\0'; i++)
        //Converting av[2][i] string into numeric and saving in "a".
    {
        a *= 10;
        a = av[2][i] - 48;
    }
    for (i = 0; av[3][i] != '\0'; i++)
        //Converting av[3][i] string into numeric and saving in "b".
    {
        b *= 10;
        b = av[3][i] - 48;
    }
    switch (av[1][0])
    {
        case '+':
            printf("%d %c %d = %d", a, av[1][0], b, a + b);
            break;
        case '-':
            printf("%d %c %d = %d", a, av[1][0], b, a - b);
            break;
        case '/':
            printf("%d %c %d = %.2f", a, av[1][0], b, float(a) / float(b));
            break;
        case '*':
            printf("%d %c %d = %d", a, av[1][0], b, a * b);

```

```

        break;
    case '%':
        printf("%d %c %d = %d", a, av[1][0], b, a % b);
        break;
    case '<':
        if (a < b)
            printf("%d %c %d is True.", a, av[1][0], b);
        else
            printf("%d %c %d is False.", a, av[1][0], b);
        break;
    case '>':
        if (a > b)
            printf("%d %c %d is True.", a, av[1][0], b);
        else
            printf("%d %c %d is False.", a, av[1][0], b);
        break;
    case '=':
        if (a == b)
            printf("%d %c %d is True.", a, av[1][0], b);
        else
            printf("%d %c %d is False.", a, av[1][0], b);
        break;
    default:
        printf("Operator '%c' is not found.\n", av[1][0]);
    }
    _getch();
    return 0;
}

```

<https://github.com/hiverkiya/Let-Us-C-Solutions>

Operations on Bits

14 July 2020 16:15

[Chapter 21: Operations On Bits] Solutions

[A]

(a)

```
#include<stdio.h>
#include<conio.h>
#define _BV(x) 1<<x
int main()
{
    unsigned int color, j, k, andmask;
    int i;
    printf("\nEnter the number : ");
    scanf("%u", &color);
    for (i = 0; i <= 6; i++)
    {
        j = i;
        andmask = _BV(j);
        puts("");
        k = color & andmask;
        if (k > 0)
        {
            switch (i)
            {
                case 0:
                    printf("Voilet, ");
                    break;
                case 1:
                    printf("Indigo, ");
                    break;
                case 2:
                    printf("Blue, ");
                    break;
                case 3:
                    printf("Green, ");
                    break;
                case 4:
                    printf("Yellow, ");
                    break;
                case 5:
                    printf("Orange, ");
                    break;
                case 6:
                    printf("Red, ");
                    break;
                default:
                    break;
            }
        }
    }
    _getch();
    return 0;
}
```

(b)

```
#include<stdio.h>
#include<conio.h>
```

```

#include<windows.h>
#include<math.h>
#define _BV(x) 1<<x
//Binary To Decimal
int bintodec(int *);
void initinfo(int *);
void addinfo(int*);
int main()
{
int i, information[10], eng_daily, up_tabloid, regional;
eng_daily = up_tabloid = regional = 0;
unsigned int andmask, j;
for (i = 0; i <= 9; i++)
{
printf("\nRespondent number %d\n", i + 1);
addinfo(&information[i]);
// 000100100 (people that reads english(3) daily(6),)
andmask = _BV(2) | _BV(5);
j = information[i] & andmask;
if (j == andmask)
eng_daily++;
// 100000001 (people with Upper(0) class and tabloid(8))
andmask = _BV(0) | _BV(8);
j = information[i] & andmask;
if (j == andmask)
up_tabloid++;
// 000001000 (people that reads in regional(5))
andmask = _BV(3);
j = information[i] & andmask;
if (j == andmask)
regional++;
}
printf("\nStatistical Data :-\n");
printf("\nPersons read English newspaper : %d", eng_daily);
printf("\nPersons belongs to Upper class and read Tabloid are : %
d", up_tabloid);
printf("\nPersons read newspaper in regional language are : %
d", regional);
_getch();
return 0;
}
int bintodec(int *num)
{
int i, dec = 0;
for (i = 0; i <= 8; i++)
dec += num[i] * (int)pow(2, 8 - i);
return dec;
}
void initinfo(int *information)
{
int i;
for (i = 0; i <= 8; i++)
information[i] = 0;
}
void addinfo(int *information)
{
int choice, info[9];
printf("\nChoose your economical class\n");
printf("1. Upper Class\n2. Middle Class\n");
printf("3. Lower Class\n");
scanf("%d", &choice);
initinfo(info);

```

```

switch (choice)
{
case 1:
    info[0] = 1;
    break;
case 2:
    info[1] = 1;
    break;
case 3:
    info[2] = 1;
default:
    break;
}
printf("\nChoose your language\n");
printf("1. English\n2. Hindi\n3. Regional\n");
scanf("%d", &choice);
switch (choice)
{
case 1:
    info[3] = 1;
    break;
case 2:
    info[4] = 1;
    break;
case 3:
    info[5] = 1;
    break;
default:
    break;
}
printf("\nChoose your newspaper category\n");
printf("\n1. Daily\n2. Supplement\n3. Tabloid\n");
scanf("%d", &choice);
switch (choice)
{
case 1:
    info[6] = 1;
    break;
case 2:
    info[7] = 1;
    break;
case 3:
    info[8] = 1;
    break;
default:
    break;
}
*information = bintodec(info);
system("cls");
}

```

(c)

```

#include<stdio.h>
#include<conio.h>
#include<windows.h>
#include<math.h>
#define _BV(x) 1<<x
int bintodec(int*);
void initinfo(int*);
void addinfo(int *);
int main()
{
int game, i, count = 0;

```

```

unsigned int andmask, j;
addinfo(&game);
for (i = 0; i <= 8; i++)
{
    andmask = _BV(i);
    j = game & andmask;
    if (j == andmask)
        count++;
}
if (count >= 5)
    printf("\nYou are eligible for champions trophy.\n");
else
    printf("\nYou aren't eligible for champions trophy.\n");
_getch();
return 0;
}
int bintodec(int *num)
{
    int i, dec = 0;
    for (i = 0; i <= 8; i++)
        dec += num[i] * (int)pow(2, 8 - i);
    return dec;
}
void initinfo(int *information)
{
    int i;
    for (i = 0; i <= 8; i++)
        information[i] = 0;
}
void addinfo(int *game)
{
    int info[9], i;
    printf("1. Cricket\n2. Basketball\n");
    printf("3. Football\n4.Hockey\n");
    printf("5.Lawn Tennis\n6.Table Tennis\n");
    printf("7.Carom\n8.Chess\n9.Kabaddi");
    printf("\nEnter the number of winning of games.");
    printf("Answer given should be in the form of 0 or 1.");
    printf("And should be given\n one by one for each game.\n");
    for (i = 0; i <= 8; i++)
        scanf("%d", &info[i]);
    *game = bintodec(info);
}

```

(d)

18 = (10010)binary
 here last four bits represent the type of the animal so, and first represent its nutrition method.

0th bit = canine

1st bit = feline

2nd bit = cetacean

3rd bit = marsupial

4th bit = Herbivorous or carnivorous.

This is an "herbivorous" animal of "feline" family.

(e)

```

#include<stdio.h>
#include<conio.h>
#include<math.h>
#define _BV(x) (1<<x)
typedef unsigned short int Time;
void time(Time, int*, int*, int*);
int main()
{

```

```

Time t = 2081;
/*Its Binary equivalent is
00001 000001 00001
So, first bit of hr, min and sec are
1, so their value will be 2^0, i.e. 1
*/
int hr = 0, min = 0, sec = 0;
time(t, &hr, &min, &sec);
printf("Hour: %d\nMin: %d\nSec: %d\n", hr, min, sec);
return 0;
}
void time(Time t, int *hr, int *min, int *sec)
{
    int i;
    for (i = 0; i < 16; i++)
    {
        if (t & _BV(i))
        {
            if (i < 5)
                *sec += int(pow(2, i));
            else if (i > 10)
                *hr += int(pow(2, i - 11));
            else
                *min += int(pow(2, i - 5));
        }
    }
}
(f)
#include<stdio.h>
#include<conio.h>
#include<math.h>
#define _BV(x) (1<<x)
void showinfo(int);
int main()
{
    int rnum, search;
    int data[] = { 273, 548, 786, 1096 };
    printf("\nEnter the room number : ");
    scanf("%d", &rnum);
    for (int i = 0; i < 4; i++)
    {
        search = 0;
        for (int j = 8; j < 16; j++)
            if (data[i] & _BV(j))
                search += int(pow(2, j - 8));
        if (rnum == search)
        {
            showinfo(data[i]);
            break;
        }
    }
    _getch();
    return 0;
}
void showinfo(int a)
{
    int search = 0;
    for (int i = 0; i < 16; i++)
    {
        //If a non zero bit is encountered
        if (a & _BV(i))
        {

```

```

if (i < 4)
{
    printf("\n\nYear: ");
    switch(i)
    {
        case 0:
            printf("First\n");
            break;
        case 1:
            printf("Second\n");
            break;
        case 2:
            printf("Thrid\n");
            break;
        case 3:
            printf("Fourth\n");
            default;;
        }
    }
else if (i > 7)
    /*calculating room number*/
    search += int(pow(2, i - 8));
else
{
    printf("Branch: ");
    switch (i)
    {
        case 4:
            printf("Mechanical\n");
            break;
        case 5:
            printf("Chemical\n");
            break;
        case 6:
            printf("Electronics\n");
            break;
        case 7:
            printf("IT\n");
            default;;
        }
    }
}
printf("Room Number: %d\n", search);
}

```

(g)

```

k = 35    l = -36 m = 0
n = 97    o = 260 p = 1

```

[B]

(a)

```

0101 1010
5      A
1100 0011
C      3
1010 1010 0111 0101
A      A      7      5
1111 0000 0101 1010
F      0      5      A

```

(b)

que: $a = a|3$


```

ans: a |= 3
que: a = a & 0x48
ans: a &= 0x48
que: b = b^0x22
ans: b ^= 0x22
que: c = c << 2
ans: c <<= 2
que: d = d >> 4
ans: d >>= 4

```

(c)

```

#include<stdio.h>
#include<conio.h>
#include<Windows.h>
#include<math.h>
#define _BV(x) 1<<x
int checkbits(int, int, int);
int main()
{
    int status;
    status = checkbits(14, 3, 3);
    if (status)
        printf("Required bits are ON\n");
    else
        printf("Required bits are OFF\n");
    _getch();
    return 0;
}
int checkbits(int x, int p, int n)
{
    unsigned int andmask = 0;
    int i, j;
    for (i = 0, j = p; i < n; i++, j--)
        andmask |= _BV(j);
    if ((x & andmask) == andmask)
        return 1;
    else
        return 0;
}

```

(d)

```

#include<stdio.h>
#include<conio.h>
#include<windows.h>
#include<math.h>
#define _BV(x) 1<<x
int checkbits(unsigned char num);
int main()
{
    int status;
    status = checkbits(200);
    if (status)
        printf("Required bits are ON\n");
    else
        printf("Required bits are OFF\n");
    _getch();
    return 0;
}
int checkbits(unsigned char num)
{
    unsigned char andmask;
    andmask = _BV(7) | _BV(6) | _BV(3);
}

```

```

    if ((num & andmask) == andmask)
        return 1;
    else
        return 0;
}
(e)

```

```

#include<stdio.h>
#include<conio.h>
#include<Windows.h>
#include<math.h>
#define _bv(x) 1<<x
#define _ls(x,y) x<<y
#define _rs(x,y) x>>y
typedef unsigned short int Store;
void bitexchange(Store*);
int main()
{
    Store num;
    printf("Enter number : ");
    scanf("%hu", &num);
    printf("Before exchange\n");
    printf("num : %u\n", num);
    bitexchange(&num);
    printf("After Exchange its bytes :-\n");
    printf("num : %u\n", num);
    _getch();
    return 0;
}
void bitexchange(Store *num)
{
    unsigned char left, right;
    // First byte (8-bits) goes to left side.
    left = _ls(*num, 8);
    //Second byts(8 - bits) goes to right side.
    right = _rs(*num, 8);
    *num = 0;
    *num = right | left;
}
(f)

```

```

#include<stdio.h>
#include<conio.h>
#include<windows.h>
#include<math.h>
#define _bv(x) 1<<x
#define _ls(x,y) x<<y
#define _rs(x,y) x>>y
typedef unsigned char Bit;
void bitexchange(Bit*);
int main()
{
    Bit num;
    printf("\nEnter the number : ");
    scanf("%hhd", &num);
    printf("\nBefore shifting\n");
    printf("\nnum : %d", num);
    bitexchange(&num);
    printf("\nAfter exchange");
    printf("\nnum : %d\n", num);
    _getch();
    return 0;
}

```

```

}
void bitexchange(Bit *num)
{
    Bit left, right;
    left = _ls(*num, 4);
    right = _rs(*num, 4);
    *num = 0;
    *num = right | left;
}

```

(g)

```

#include<stdio.h>
#include<conio.h>
#include<windows.h>
#include<math.h>
#define _BV(x) 1<<x
typedef unsigned char Bit;
void oddbiton(Bit*);
int main()
{
    Bit num;
    printf("Enter the number : ");
    scanf("%hhd", &num);
    oddbiton(&num);
    _getch();
    return 0;
}
void oddbiton(Bit *num)
{
    Bit andmask = 0;
    int i;
    for (i = 1; i < 8; i += 2)
        andmask |= _BV(i);
    *num |= andmask;
}

```

(h)

```

#include<stdio.h>
#include<conio.h>
#include<windows.h>
#include<math.h>
#define _BV(x) 1<<x
#define _ls(x,y) x<<y
#define _rs(x,y) x>>y
int main()
{
    unsigned char num, andmask = 0;
    printf("Enter the number : ");
    scanf("%hhd", &num);
    andmask = ~(_BV(3) | _BV(5));
    num &= andmask;
    _getch();
    return 0;
}

```

(i)

```

#include<stdio.h>
#include<conio.h>
#define _BV(x) 1<<x
int main()
{
    unsigned char num, andmask = 0;

```

```

printf("Enter the number : ");
scanf("%hhd", &num);
andmask = _BV(3) | _BV(5);
num |= andmask;
_getch();
return 0;
}
(j)

```

```

#include<stdio.h>
#include<conio.h>
#define _BV(x) 1<<x
void showbits(unsigned char);
int main()
{
    int i;
    i = 10;
    showbits(i);
    _getch();
    return 0;
}
void showbits(unsigned char num)
{
    int i;
    unsigned char andmask;
    for (i = 7; i >= 0; i--)
    {
        andmask = _BV(i);
        ((andmask&num) == 0) ? printf("0") : printf("1");
    }
}

```

<https://github.com/hiverkiya/Let-Us-C-Solutions>

Miscellaneous Features

14 July 2020 16:15

[Chapter 22: Miscellaneous Features] Solutions

[A]

(a) 0 1 2

(b) 1.0

(c) 4

[B]

(a) No Error

(b) There are two methods to invoke a function,
(*func)();
func();

(c) No Error

[C]

(a)

```
#include<stdio.h>
#include<conio.h>
float one(int, int);
float two(int, int);
float three(int, int);
float four(int, int);
int main()
{
    float(*ptr[4])(int, int);
    float a, b, c, d;
    ptr[0] = one;
    ptr[1] = two;
    ptr[2] = three;
    ptr[3] = four;
    a = ptr[0](1, 2);
    b = ptr[1](2, 4);
    c = ptr[2](6, 3);
    d = ptr[3](22, 7);
    printf("a : %f\tb : %f\n", a, b);
    printf("c : %f\td : %f\n", c, d);
    _getch();
    return 0;
}
float one(int a, int b)
{
    float r;
    r = (float)a / b;
    return r;
}
float two(int x, int y)
{
    float ans;
    ans = (float)x / y;
    return ans;
}
float three(int i, int j)
{
```

```

float a;
a = (float)i / j;
return a;
}
float four(int m, int n)
{
return ((float)m / n);
}

```

(b)

```

#include<stdio.h>
#include<conio.h>
#include<stdarg.h>
void type(int, ...);
int main()
{
type(5, 4, 6, 3, 6, 7);
return 0;
}
void type(int point, ...)
{
if (point < 1)
{
printf("Nothing can be draw.");
return;
}
switch (point)
{
case 1:
printf("A point can be drawn.\n");
break;
case 2:
printf("A line can be drawn.\n");
break;
case 3:
printf("A triangle can be drawn.\n");
break;
case 4:
printf("A rectangle can be drwan.\n");
break;
case 5:
printf("A pentagon can be drawn.\n");
break;
case 6:
printf("A hexagon can be drawn.\n");
break;
default:
printf("A polygon can be drawn.\n");
}
}

```

(c)

```

#include<stdio.h>
#include<conio.h>
#include<stdarg.h>
#include<Windows.h>
#define MAX 5
typedef struct date
{
unsigned d : 5;
unsigned m : 4;
unsigned y : 12;

```

```

}EMP;
void swap(EMP*, EMP*);
int main()
{
    int i, j;
    EMP e[10], a[10];
    e[0].y = 2010; e[0].m = 12; e[0].d = 10;
    e[1].y = 1990; e[1].m = 3; e[1].d = 23;
    e[2].y = 1995; e[2].m = 4; e[2].d = 13;
    e[3].y = 2001; e[3].m = 1; e[3].d = 13;
    e[4].y = 1990; e[4].m = 3; e[4].d = 20;
    e[5].y = 1992; e[5].m = 6; e[5].d = 19;
    for (i = 0; i < MAX; i++)
        a[i] = e[i];
    for (i = 0; i < MAX; i++)
    {
        for (j = i + 1; j < MAX; j++)
        {
            if (a[j].y < a[i].y)
                swap(&a[i], &a[j]);
            if (a[j].y == a[i].y)
            {
                if (a[j].m < a[i].m)
                    swap(&a[i], &a[j]);
                if (a[j].m == a[i].m)
                    if (a[j].d < a[i].d)
                        swap(&a[i], &a[j]);
            }
        }
    }
    system("cls");
    for (i = 0; i <= 4; i++)
    {
        printf("\nEmployee no. %d : ", i + 1);
        printf("%2d/%2d/%4d", a[i].d, a[i].m, a[i].y);
    }
    _getch();
    return 0;
}
void swap(EMP *a, EMP *b)
{
    EMP c;
    c = *a;
    *a = *b;
    *b = c;
}
(d)

```

```

#include<stdio.h>
#include<conio.h>
#include<Windows.h>
#define MAX 2
int main()
{
    enum sex { male, female };
    enum level { major, minor };
    typedef struct people
    {
        enum sex gender;
        enum level age_level;
        char policy_name[10];
        unsigned int duration_in_year;
    }
}

```

```

} PEOPLE;
PEOPLE p[MAX];
int i, enm;
for (i = 0; i < MAX; i++)
{
    printf("\nEnter the gender (male = 0 or female = 1) : ");
    scanf("%d", &enm);
    enm ? (p[i].gender = female) : (p[i].gender = male);
    printf("\nEnter the age status (major = 0 or minor = 1) : ");
    scanf("%d", &enm);
    enm ? (p[i].age_level = minor) : (p[i].age_level = major);
    //Clearing input stream
    while (getchar() != '\n');
    printf("\nEnter name of the customer : ");
    gets_s(p[i].policy_name);
    printf("\nEnter the duration (in years) : ");
    scanf("%d", &p[i].duration_in_year);
}
system("cls");
for (i = 0; i < MAX; i++)
{
    printf("\n%d\t%d", p[i].gender, p[i].age_level);
    printf("\t%s\t%d", p[i].policy_name, p[i].duration_in_year);
}
_getch();
return 0;
}

```

<https://github.com/hiverkiya/Let-Us-C-Solutions>

C Under Linux

14 July 2020 16:15

[Chapter 23: C Under Linux] Solutions

[A]

(a) True

(b) False (There could be some variations in various distributions)

(c) False (Generally, the scheduling unit on Linux is referred to as a KSE, a "kernel scheduling entity".
On modern Linux systems, each thread is a KSE.)

(d) False

(e) False

(f) True

(g) True

(h) False

(i) True

(j) False

(k) False

(l) False

(m) False

(n) False

(o) False

(p) True

(q) True

(r) True

(s) True

[B]

(a)

2^4 i.e, 16 Total Processes.

(b)

A process which has finished the execution but still has entry in the process table to report

to its parent process is known as a zombie process.

A process whose parent process no more exists i.e.

either finished or terminated without waiting for its child process to terminate is called an orphan process.

(c)

```
// Name of program mainreturn.cpp
#include <iostream>
using namespace std;

int main(int argc, char** argv)
{
    cout << "You have entered " << argc
        << " arguments:" << "\n";

    for (int i = 0; i < argc; ++i)
        cout << argv[i] << "\n";

    return 0;
}
```

(d)

getppid() : returns the process ID of the parent of the calling process.
If the calling process was created by the fork() function and the parent process still exists at the time of the getppid function call, this function returns the process ID of the parent process.
Otherwise, this function returns a value of 1 which is the process id for init process.
getpid() : returns the process ID of the calling process.
This is often used by routines that generate unique temporary filenames.

(e)

Can be done just by little logical analysis.

(f)

To prevent of zombie processes you need to tell the parent to wait for the child,
until the child's terminates the process.

You need to use the waitpid() function that is included in the library 'sys/wait.h'

(g)

Signal is a notification, a message sent by either operating system or some application to your program (or one of its threads).

Each signal identified by a number, from 1 to 31.

Signals don't carry any argument and their names are mostly self explanatory.

For instance SIGKILL or signal number 9 tells the program that someone tries to kill it.

(h)

Signals are used for a wide variety of purposes in Unix programming, and we've already used them in smaller contexts.

a program that handles a signal, either by ignoring it or taking some action when the signal is delivered is described as signal handler. For rest of answer prefer

[This Site](https://en.wikipedia.org/wiki/C_signal_handling)

(i)

No, you can't catch the uncatchable signals as it is always caught by the default handler implemented by the kernel.

SIGKILL always terminates the process

. Even if the process attempts to handle the SIGKILL signal by registering a handler for it,

still the control would always land in the default SIGKILL handler which w

ould terminate the program.

This is what happens when you try to shut down your system.

First, the system sends a SIGTERM signal to all the processes

and waits for a while giving those processes a grace period.

If it still doesn't stop even after the grace period,

the system forcibly terminates all the process by using SIGKILL signal.

(j)

```
#include<stdio.h>
#include<signal.h>
#include<unistd.h>
void sig_handler(int signo)
{
    if (signo == SIGINT)
printf("received SIGINT\n");
}
int main(void)
{
    if (signal(SIGINT, sig_handler) == SIG_ERR)
printf("\ncan't catch SIGINT\n");
    // A long long wait so that we can easily issue a signal to this process
    while(1)
sleep(1);
    return 0;
}
```

(k)

```
/* A C program that does not terminate when Ctrl+C is pressed */
#include <stdio.h>
#include <signal.h>

/* Signal Handler for SIGINT */
void sigintHandler(int sig_num)
{
    /* Reset handler to catch SIGINT next time.
    Refer http://en.cppreference.com/w/c/program/signal */
    signal(SIGINT, sigintHandler);
    printf("\n Cannot be terminated using Ctrl+C \n");
    fflush(stdout);
}

int main ()
{
    /* Set the SIGINT (Ctrl-C) signal handler to sigintHandler
    Refer http://en.cppreference.com/w/c/program/signal */
    signal(SIGINT, sigintHandler);

    /* Infinite loop */
    while(1)
    {
    }
    return 0;
}
```

Please Refer [[GeeksforGeeks](https://www.geeksforgeeks.org/write-a-c-program-that-doesnt-terminate-when-ctrlc-is-pressed/)](<https://www.geeksforgeeks.org/write-a-c-program-that-doesnt-terminate-when-ctrlc-is-pressed/>) and [[Quora](https://www.quora.com/What-is-the-difference-between-the-SIGINT-and-SIGTERM-signals-in-Linux-What%E2%80%99s-the-difference-between-the-SIGKILL-and-SIGSTOP-signals)]

(<https://www.quora.com/What-is-the-difference-between-the-SIGINT-and-SIGTERM-signals-in-Linux-What%E2%80%99s-the-difference-between-the-SIGKILL-and-SIGSTOP-signals>) for much brief explanations.

<https://github.com/hiverkiya/Let-Us-C-Solutions>

Periodic Tests

14 July 2020 16:16

[Periodic Test -1] Solutions

[A]

- (1) i+1
- (2) Real
- (3) ; (Statement Terminator)
- (4) 2 or 4 bytes
- (5) Do-while

[B]

- (1) True
- (2) True
- (3) False
- (4) True
- (5) False

[C]

- (a) 5 5 6
- (b) 65 A
- (c) 2 1
- (d) 21525
- (e) 0 1

[D]

- (a) Syntax of logical connectives is incorrect.
- (b) "then" is not any type of syntax in C language.
- (c) Parentheses should be used in conditional operators to specify the associativity of operations.
- (d) NO ERROR
- (e) NO ERROR

[E]

(1)

```
#include <stdio.h>
int fact (int);
int main()
{
    long int sum=0;
    int i;
    for(i=1;i<=9;i++)
    {
```

```

        sum+=(fact(i)*fact(i+1));
    }
    printf("%ld",sum);
    return 0;
}
int fact(int fact)
{
    int result=1,i;
    for(i=1;i<=fact;i++)
    {
        result*=i;
    }
    return result;
}

```

(2)

```

#include<stdio.h>
int main()
{
    int cp=0,cn=0,cz=0,number;
    char another='y';
    while(another=='y')
    {
        scanf("%d",&number);

        if(number<0)
        {
            cn++;
        }
        if(number>0)
        {
            cp++;
        }
        if(number==0)
        {
            cz++;
        }

        printf("Enter Another Number y/n ?\n");
        fflush (stdin);
        scanf(" %c",&another);
    }
    printf("Number of Positives = %d\n",cp);
    printf("Number of Negatives = %d\n",cn);
    printf("Number of Zeroes = %d\n",cz);
    return 0;
}

```

(3)

```

#include<stdio.h>
int main()
{
    int size,i;
    printf("Enter number of elements you want to Enter\n");
    scanf("%d",&size);
    int array[size];
    printf("Enter Elements now \n");
    for(i=0;i<size;i++)
    {
        scanf("%d",&array[i]);
    }
}

```

```

int large=array[0],small=array[0];
for(i=1;i<size;i++)
{
    if(array[i]>large)
    {
        large=array[i];
    }
    if(array[i]<small)
    {
        small=array[i];
    }
}
printf("The Largest Number is %d and Smallest Number is %d.\n",large,small);
printf("The Range is %d",large-small);
return 0;
}

```

(4)

```

#include<stdio.h>
int main()
{
    int num1,num2,num3;
    printf("Enter Numbers one by one.\n");
    scanf("%d %d %d",&num1,&num2,&num3);
    int smallest,largest,sum=0,middle;
    sum=num1+num2+num3;
    smallest=num1<num2?(num1<num3?num1:num3):(num2<num3?num2:num3);
    largest=num1>num2?(num1>num3?num1:num3):(num2>num3?num2:num3);
    middle=sum-smallest-largest;
    if((largest*largest)==(smallest*smallest)+(middle*middle))
    {
        printf("Entered Numbers are Pythagorean triplet.\n");
    }
    else
    {
        printf("Entered Numbers are not Pythagorean triplet.\n");
    }
    return 0;
}

```

[Periodic Test -2] Solutions

[A]

- (1) clrscr()
- (2) Pointers
- (3) & (Ampersand)
- (4) Macro Expansion
- (5) Address

[B]

(1) False

(2) True

(3) True

(4) True

(5) False

[C]

(1)

The stack can be easily allocated as well as freed again, so it is a natural choice.

All the variable addresses are relative to the stack pointer that's incremented at each call or return.

Fast easy way to allocate and cleanup memory used by these variables.

(2)

When we want to make changes to original value's we use "call by reference" ,

other way if we want to work on some temporary copy of original values we use "call by value".

(3)

Sizeof a pointer is independent of type of the variables because the type of

every pointer variable is implicitly unsigned integer.

(4)

Hole concept is found in the structures in c language where the compiler leaves certain holes

in the memory alignment when it allocates memory on the page boundaries.

That's the reason you get the sizeof structure some times greater than the sum of all the fields in the structure .

Also called structure padding ,structure padding can be avoided by using a preprocessor

directive #pragma 1 the compiler will allocate memory in multiples of one.

(5)

Stopping condition can be any conditional statement (most cases), or it can be exception depending upon the implementation.

Any recursive function without stoppage condition goes to infinite loop and blow up the stack.

[D]

(1)

```
#include<stdio.h>
int fun(int,int,int,int);
int main()
{
printf("Enter four integers\n");
int a,b,c,d;
scanf("%d %d %d %d",&a,&b,&c,&d);
fun(a,b,c,d);
return 0;
}
int fun(int a,int b,int c,int d)
```

```

{
printf("Sum is %d.\n",a+b+c+d);
printf("Product is %d.\n",a*b*c*d);
printf("Average is %f.\n",(float)(a+b+c+d)/4.0);
return 0;
}

```

(2)

```

int func(int);
#include <stdio.h>
int main()
{   int number;
printf("Enter Number\n");
scanf("%d",&number);
func(number);
return 0;
}
int func(int num)
{
int counter_var;
for(counter_var=2;counter_var<=num;counter_var++)
{
    if(num%counter_var==0)
    {
        printf("%d\n",counter_var);
        func(num/counter_var);
        break;
    }
}
return 0;
}

```

(3)

```

#define PI 3.14159265
#include<stdio.h>
int main()
{   float radius;
printf("Enter radius\n");
scanf("%f",&radius);
printf("The area of circle is %f.\n",PI*radius*radius);
printf("The circumference of circle is %f.\n",2*PI*radius);
printf("Volume of sphere is %f.\n",4.0/3.0*PI*radius*radius*radius);
printf("Enter height of cone\n");
float height;
scanf("%f",&height);
printf("Volume of cone is %f.\n",PI*radius*radius*(height/3.0));
return 0;
}

```

(4)

```

#include <stdio.h>
int main()
{
int integerType;
float floatType;
double doubleType;
char charType;
// Sizeof operator is used to evaluate the size of a variable
printf("Size of int: %ld bytes\n",sizeof(integerType));
}

```



```

printf("Size of float: %ld bytes\n",sizeof(floatType));
printf("Size of double: %ld bytes\n",sizeof(doubleType));
printf("Size of char: %ld byte\n",sizeof(charType));
return 0;
}

```

[Periodic Test -3] Solutions

[A]

- (1) Base Address
- (2) Index and Size
- (3) number
- (4) 0
- (5) different

[B]

- (1) False
- (2) True
- (3) True
- (4) True
- (5) True

[C]

- (1) It Overwrite the next elements present in the memory.
- (2) When we want to know information about every element in an array .
This can just only be done with help of structures.
Example: If we have elemnts of char type then the characters are
in a sequence that we are unable to find the correct info.
but with help of structure we can easily distinguish that and easily g
et to know about the name
Suppose we made an array VIKASKUMAR.
Here we dont know that Vikaskumar is one person or Vikas is one and ku
mar is another it can only
be Identified with the help of structures.
- (3) The limitation is that we cant just declare an array of pointers.
We can overcome this by initialiing an array of pointers along with its de
claration.
- (4) In 2D array a[4][4] is an array.
Mentioning the array name in C or C++ gives the base address in all contex
ts except one.
Syntactically, the compiler treats the array name as a pointer to the
first element.
You can reference elements using array
syntax, a[n], or using pointer syntax, *(a+n),
and you can even mix the usages within an expression.

(5) For receiving multiwords strings using scanf we can write it as
scanf("%[^\n]
s,stringname) in some compilers while others accept normally scanf("%
s",stringname).
gets(stringname).

[D]

SOLVED ALREADY IN PREVIOUS CHAPTERS OR SIMILAR CODES ARE AVAILABLE.

[Periodic Test-4] Solutions

[A]

- (1) Hexadecimal values
- (2) Integers
- (3) Signal Handler
- (4) sigprocmask;
- (5) Complement

[B]

- (1) False
- (2) True
- (3) False
- (4) True
- (5) True

[C]

THESE QUESTION'S ANSWERS ARE EASILY AVAILABLE ON INTERNET.

[D]

(1)

```
#include<stdio.h>
main()
{
int a, b, result;
printf("\nEnter the numbers to be multiplied:");
scanf("%d%d", &a, &b);          // a > b
result = 0;
while (b != 0)                  // Iterate the loop till b == 0
{
    if (b & 01)                  // Bitwise & of the value of b with 01
    {
        result = result + a;    // Add a to result if b is odd .
    }
    a<<=
1;                               // Left shifting the value contained in 'a' by 1
                                // Multiplies a by 2 for each loop
    b>>=
```

```

1;          // Right shifting the value contained in 'b' by 1.
    }
    printf("nResult:%d",result);
    }

```

```

(2)
#include<stdio.h>
#include<conio.h>
#include<ctype.h>
void main()
{
    FILE *f;
    char ch;
    int line=0,word=0;
    clrscr();
    f=fopen("student","w");
    printf("Enter text press ctrl+z to quit\n");
    do
    {
        ch=getchar();
        putc(ch,f);
    }
    while(ch!=EOF);
    fclose(f);
    f=fopen("student","r");
    while((ch=getc(f))!=EOF)
    {
        if(ch=='\n')
            line++;
        if(isspace(ch)||ch=='\t'||ch=='\n')
            word++;
        putchar(ch);
    }
    fclose(f);
    printf("\n no of line=%d\n",line);
    printf("no of word=%d\n",word);
    getch();
}

```

```

(3)
SIMILAR CODE AVAILABLE IN CHAPTER RELATED TO THIS TOPIC.

```

```

(4)

#include<stdio.h>

int main() {
FILE *fp1, *fp2;
int ch1, ch2;
char fname1[40], fname2[40];

printf("Enter name of first file :");
gets(fname1);

printf("Enter name of second file:");
gets(fname2);

fp1 = fopen(fname1, "r");
fp2 = fopen(fname2, "r");

if (fp1 == NULL) {
    printf("Cannot open %s for reading ", fname1);
    exit(1);
} else if (fp2 == NULL) {
    printf("Cannot open %s for reading ", fname2);
}
}

```

```
exit(1);
} else {
ch1 = getc(fp1);
ch2 = getc(fp2);

while ((ch1 != EOF) && (ch2 != EOF) && (ch1 == ch2)) {
    ch1 = getc(fp1);
    ch2 = getc(fp2);
}

if (ch1 == ch2)
    printf("Files are identical n");
else if (ch1 != ch2)
    printf("Files are Not identical n");

fclose(fp1);
fclose(fp2);
}
return (0);
}
```

<https://github.com/hiverkiya/Let-Us-C-Solutions>