1. **Assignments on Variables, Operators, Expressions and Assignment statements**

**1. Define an integer (int) variable. Read a number from the user into this variable. Finally print the value in the variable. Also print the variable as hexadecimal(hex) number and as octal number.**

Ans:

#include <stdio.h>

int main()

{

int num;

printf("Enter the number:\n");

scanf("%d",&num);

printf("In Decimal =%d\n",num);

printf("In Hexadecimal =%x\n",num);

printf("In Octal =%o\n",num);

return 0;

}

**2. Define a short integer (short) variable. Read a number from the user into this variable. Finally print the value in the variable. Also print the variable as hex number and octal number.**

Ans:

#include <stdio.h>

int main()

{

short int num;

printf("Enter the number:\n");

scanf("%hd",&num);

printf("In Decimal =%hd\n",num);

printf("In Hexadecimal =%hx\n",num);

printf("In Octal =%ho\n",num);

return 0;

}

**3. Define a character (char) variable. Read a character from the user into this variable. Finally print the character in the variable. Also print the char variable as decimal number and also as hex number.**

Ans:

#include <stdio.h>

int main()

{

char a;

printf("Enter the varible:\n");

scanf("%c",&a);

printf("You Enter varible: %c",a);

printf("In Decimal =%d\n",a);

printf("In Hexadecimal =%x\n",a);

return 0;

}

**4. Define a small integer (char) variable. Read a small number from the user into this variable. Finally print the variable. Also print the variable as hex number and octal number.**

Ans:

#include <stdio.h>

int main()

{

short int a;

printf("Enter the varible:\n");

scanf("%hd",&a);

printf("You Enter varible: %hd\n",a);

printf("In Hexadecimal =%x\n",a);

printf("In Octal =%o\n",a);

return 0;

}

**5. Define a 64 bit integer (long long) variable. Ask user to enter his mobile number. Read and store the mobile number in the 64 bit integer. Print the mobile phone number.**

Ans:

#include <stdio.h>

int main()

{

long long int num;

printf("Enter the mobile number:\n");

scanf("%lld",&num);

printf("You Enter mobile phone number :%lld\n",num);

return 0;

}

**6. Define two integer variables. Read two numbers from the user into these variables. Next calculate and print the sum, difference, product, quotient and remainder.**

Ans:

#include <stdio.h>

int main()

{

int num1,num2,sum,sub,mul,divi,mod;

printf("Enter the two number:\n");

scanf("%d%d",&num1,&num2);

sum=num1+num2;

sub=num1-num2;

mul=num1\*num2;

divi=num1/num2;

mod=num1%num2;

printf("Sum of two number :%d\n",sum);

printf("difference of two number :%d\n",sub);

printf("product of two number :%d\n",mul);

printf("quotient of two number :%d\n",divi);

printf("remainder of two number :%d\n",mod);

return 0;

}

**7. Read two integers from the user (that is first define two integer variables and then read). Perform all possible conditional operations on these two integers. Print the results of all these conditional operations. Following are the possible conditinal operators:**

**Equal to(==) Less than(<) Greater than(>) Less than or Equal to (<=) Greater than or equal to(>=) Not equal to(!=)**

Ans:

#include <stdio.h>

int main()

{

int num1,num2,res;

printf("Enter the two number:\n");

scanf("%d%d",&num1,&num2);

if(num1==num2)

printf("Num1 %d is equal to num2 %d\n",num1,num2);

if(num1<num2)

printf("Num1 %d is less than num2 %d\n",num1,num2);

if(num1<=num2)

printf("Num1 %d is less than equal to num2 %d\n",num1,num2);

if(num1>num2)

printf("Num1 %d is greater than num2 %d\n",num1,num2);

if(num1>=num2)

printf("Num1 %d is greater than equal to num2 %d\n",num1,num2);

if(num1!=num2)

printf("Num1 %d is not equal to num2 %d\n",num1,num2);

return 0;

}

**8. Read two small integers from the user in hex format. You should ask user to enter only hexadecimal numbers with maximum two digits. Perform all possible bit wise operations on these two numbers. Print the results of all these bit wise operations. Following are the possible bit wise operations:**

**AND(&) OR(|) EXOR(^) Complement(~) Shift left(<<) Shift right (>>)**

Ans:

#include <stdio.h>

int main()

{

short int num1,num2,res;

printf("Enter only hexadecimal numbers with maximum two digits:\n");

scanf("%hx%hx",&num1,&num2);

res=num1&num2;

printf("%d & %d =%d \n",num1,num2,res);

res=num1|num2;

printf("%d | %d =%d \n",num1,num2,res);

res=num1^num2;

printf("%d ^ %d =%d \n",num1,num2,res);

res= ~num1;

printf("Complement %d is %d \n",num1,res);

res= ~num2;

printf("Complement %d is %d \n",num2,res);

res=num1<<num2;

printf("%d << %d =%d \n",num1,num2,res);

res=num1>>num2;

printf("%d >> %d =%d \n",num1,num2,res);

return 0;

}

**9. Read two integers from the user. Perform all possible logical operations on these two integers. Print the results of all these logical operations. Following are the possible logical operators:**

AND(&&) OR(||) NOT(!)

Ans:

#include <stdio.h>

int main()

{

int num1,num2,res;

printf("Enter two number:\n");

scanf("%d%d",&num1,&num2);

res=num1&&num2;

printf("%d && %d =%d \n",num1,num2,res);

res=num1||num2;

printf("%d || %d =%d \n",num1,num2,res);

res= !num1;

printf("%d logical not is %d \n",num1,res);

res= !num2;

printf("%d logical not is %d \n",num2,res);

return 0;

}

**10. Read three integers a, b, and c from the user.**

**Write the condition to check if a, b and c are equal.**

**Write the condition to check a is bigger than b and c.**

**Write the condition to check a is smaller than b and c.**

**Print the condition result in each case.**

**Note: Use logical operators to combine the conditional operations on these**

**variables.**

Ans:

#include <stdio.h>

int main()

{

int a,b,c,res;

printf("Enter a,b,c value:\n");

scanf("%d%d%d",&a,&b,&c);

if(a==b && a==c)

printf("a,b,c are equal\n");

if(a>b && a>c)

printf(" a is bigger than b and c\n");

if(a<b && a<c)

printf("a is smaller than b and c\n");

return 0;

}

**11. Define two float variables, read two float numbers from the user. Print their sum, difference, product and quotient.**

Ans:

#include <stdio.h>

int main()

{

float num1,num2,res;

printf("Enter two float varible:\n");

scanf("%f%f",&num1,&num2);

res=num1+num2;

printf("sum of %.2f + %.2f =%0.2f\n",num1,num2,res);

res=num1-num2;

printf("difference of %.2f - %.2f =%0.2f\n",num1,num2,res);

res=num1\*num2;

printf("product of %.2f \* %.2f =%0.2f\n",num1,num2,res);

res=num1/num2;

printf("quotient of %.2f / %.2f =%0.2f\n",num1,num2,res);

return 0;

}

**12. Define two double variables, read two float numbers from the user. Print their sum, difference, product and quotient.**

Ans:

#include <stdio.h>

int main()

{

double num1,num2,res;

printf("Enter two float varible:\n");

scanf("%lf%lf",&num1,&num2);

res=num1+num2;

printf("sum of %.2lf + %.2lf =%0.2lf\n",num1,num2,res);

res=num1-num2;

printf("difference of %.2lf - %.2lf =%0.2lf\n",num1,num2,res);

res=num1\*num2;

printf("product of %.2lf \* %.2lf =%0.2lf\n",num1,num2,res);

res=num1/num2;

printf("quotient of %.2lf / %.2lf =%0.2lf\n",num1,num2,res);

return 0;

}

**13. Define two double variables with names as 'degrees' and 'radians'. From the user read degrees. Convert the degrees to radians and store in the 'radians' variable. Finally print the degrees and radians.**

Ans:

#include <stdio.h>

#define pi 3.14

int main()

{

double degrees,radians;

printf("Enter degrees:\n");

scanf("%lf",&degrees);

radians=degrees\*(pi/180);

printf("degrees %lf and radians %lf \n",degrees,radians);

return 0;

}

**14. Define a double variable with name as 'angle'. Read degrees from the user into this 'angle'. Convert 'angle' into radians.**

Ans:

#include <stdio.h>

#define pi 3.14

int main()

{

double angle,radian;

printf("Enter degrees:");

scanf("%lf",&angle);

radian=angle\*(pi/180);

printf("angle in degree %.2lf\n",angle);

printf("angle in radian %.2lf\n",radian);

return 0;

}

**15. Read angle in degrees from the user into a double variable. Convert the degrees to radians. Use math library function sin() and calculate the sine of given angle. Print the degrees given, radians and its sign value.**

Ans:

#include <stdio.h>

#include <math.h>

#define pi 3.14

int main()

{

double angle,radian,sine;

printf("Enter angle in degrees:\n");

scanf("%lf",&angle);

radian=angle\*(pi/180);

sine = sin(radian);

printf("Degrees: %.2lf\n",angle);

printf("Radians: %.2lf\n",radian);

printf("Sine: %.2lf\n",sine);

return 0;

}

**16. Extend the above program to calculate the cos and tan values for the given degrees by using math library functions cos() and tan().**

Ans:

#include <stdio.h>

#include <math.h>

#define pi 3.14

int main()

{

double angle,radian,sine,cosin,tans;

printf("Enter angle in degrees:\n");

scanf("%lf",&angle);

radian=angle\*(pi/180);

sine = sin(radian);

cosin = cos(radian);

tans = tan(radian);

printf("Degrees: %.2lf\n",angle);

printf("Radians: %.2lf\n",radian);

printf("Sin: %.2lf\n",sine);

printf("Cosine: %.2lf\n", cosin);

printf("Tangent: %.2lf\n", tans);

return 0;

}

**17. Extend the above program to convert sin, cos and tan values back to radians by using inverse functions of sin, cos and tan.**

Ans:

#include <stdio.h>

#include <math.h>

#define pi 3.14

int main()

{

double angle,radian,sine,cosin,tans,sinradian,cosradian,tanradian;

printf("Enter angle in degrees:\n");

scanf("%lf",&angle);

radian=angle\*(pi/180);

sine = sin(radian);

cosin = cos(radian);

tans = tan(radian);

sinradian = asin(sine);

cosradian = acos(cosin);

tanradian = atan(tans);

printf("Degrees: %.2lf\n",angle);

printf("Radians: %.2lf\n",radian);

printf("Sine: %.2lf\n", sine);

printf("Sine in Radians: %.2lf\n",sinradian);

printf("Cosine: %.2lf\n", cosin);

printf("Cosine in Radians: %.2lf\n",cosradian);

printf("Tangent: %.2lf\n", tans);

printf("Tangent in Radians: %.2lf\n",tanradian);

return 0;

}

**Q18. Read a double value from the user, calculate its square root by using**

**math library function sqrt(). Print the given number and its square root.**

#include <stdio.h>

#include <math.h>

int main()

{

double num,sqroot;

printf("enter the number\n");

scanf("%lf", &num);

sqroot=sqrt(num);

printf("the given number is %.2lf\n",num);

printf("the sqroot of the given number is %.2lf\n",sqroot);

return 0;

}

**Q19. Write a program to use the math function pow(). Read base and power**

**values fron the user as double variables. Use pow() to calculate the**

**base raised by the power. Finally print the base, power and base raised**

**by power.**

\*/

#include <stdio.h>

#include <math.h>

int main() {

double base, power, result;

printf("Enter the base: ");

scanf("%lf", &base);

printf("Enter the power: ");

scanf("%lf", &power);

result = pow(base, power);

printf("Base: %.2lf\n", base);

printf("Power: %.2lf\n", power);

printf("Result: %.2lf\n", result);

return 0;

}

**Q20.Read angle in degrees as double variable. Calculate sine and cos of     given angle. Compute square root of , sum of sin square and cos square.     For any angle this value should be 1.**

#include<stdio.h>

#include<math.h>

int main()

{

double angle,sine, cose;

printf("enter the angle in degree\n");

scanf("%lf", &angle);

sine=sin(angle);

cose=cos(angle);

int sum=sqrt((sine\*sine)+(cose\*cose));

printf("angle: %lf\n", angle);

printf("sine: %lf\n", sine);

printf("cose: %lf\n", cose);

printf("sum of sqrootof of sine square & cossquare %d\n",sum);

}

**Q21.  Read the radius of a sphere from the user. Calculate the volume of the     sphere and print it. Use all variables of type 'double' only.     Formula for volume of sphere is : (4.0/3.0) \* PI \* r \* r \* r**

#include<stdio.h>

#define PI 3.14

int main()

{

double rad, vol;

printf("enter the radius of the sphere\n");

scanf("%lf", &rad);

vol= (4.0/3.0)\*PI\*rad\*rad\*rad;

printf("vol of the sphere is %.2lf\n", vol);

}

**Q22.  Read the radius of a sphere from the user. Calculate the surface are     of the sphere and print it. Use all variables of type 'double' only.     Formula for surface area of sphere is : 4.0 \* PI \* r \* r**

#include<stdio.h>

#define PI 3.14

int main()

{

double rad, surface\_area;

printf("enter the radius of the sphere\n");

scanf("%lf", &rad);

surface\_area= 4.0\*PI\*rad\*rad;

printf("vol of the sphere is %.2lf\n", surface\_area);

}

**Q23.Read the radius of sphere from the user. Ask user to enter the radius in     the units of centi meters only. Print the values of densities to the     user as given below**

Following are the densities of some metals:

  Aluminum       2.712 grams/cm3 (grams per centimeter cube)

  Copper         8.940

Gold          19.320

  Iron           7.850

   Silver        10.490

**2.   Next ask the user to enter the density of the metal of the sphere.      Finally print the weight of the sphere.     Formula for weight : volume \* Density.**

#include<stdio.h>

#define PI 3.14

int main()

{

double radius, vol, weight,density;

double aluminum= 2.712;

double copper= 8.940;

double gold= 19.320;

double iron= 7.850;

double silver= 10.490;

printf("Enter the radius in units of centimeters:\n");

scanf("%lf", &radius);

//radius=radius/100;

vol= (4.0/3.0)\*PI\*radius\*radius\*radius;

/\* printf("Following area the densities of some metal:\n");

printf("Aluminum: %.3lf\n", aluminum\*vol);

printf("Copper: %.3lf\n", copper\*vol);

printf("Gold: %.3lf\n", gold\*vol);

printf("Iron: %.3lf\n", iron\*vol);

printf("Silver: %.3lf\n", silver\*vol);\*/

printf("Following area the densities of some metal:\n");

printf("Aluminum 2.712 grams/cm3\n");

printf("Copper 8.940 grams/cm3\n");

printf("Gold 19.320 grams/cm3\n");

printf("Iron 7.850 grams/cm3\n");

printf("Silver 10.490 grams/cm3\n");

printf("enter the density of metal in gram /cm3:\n");

scanf("%lf", &density);

weight= vol\*density;

printf("the weight of the density is %3lf\n", weight);

return 0;

}

**Q24. Following is the formula for the compound interest**

**A = P \* ( (1 + (r/n)) \*\* (n\*t) )**

**Note: \*\* means 'Power of'**

**P = Principal amount (the initial amount you borrow or deposit)**

**r = Annual rate of interest**

**t = Number of years the amount is deposited or borrowed for**

**A = Amount of money accumulated after n years, including interest**

**n = Number of times the interest is compounded per year**

**Write a program which reads P,r,n and t. Use all variables of type**

**double only. Compute the value of 'A' and print it.**

**Hint: Use 'power of' function pow()**

#include<math.h>

int main()

{

double A, P,r,n,t;

printf("Enter the Principal amount:\n");

scanf("%lf", &P);

printf("Enter the amount rate of intrest:\n");

scanf("%lf", &r);

printf("Enter the number of times the intrest is compund per year:\n");

scanf("%lf", &n);

printf("Enter the number of years the amount is deposited or borrow for:\n");

scanf("%lf", &t);

A= P\* pow((1+(r/n)),(n\*t));

printf("the amount of money accumulated after %0.2lf years is : %0.2lf\n",t,A);

return 0;

}

**Q25.Read a character into char variable and print this as small integer in**

**decimal and hex.**

#include<stdio.h>

int main()

{

char ch;

printf("enter the character\n");

scanf("%c", &ch);

printf("In Decimal: %hd\n",ch);

printf("In Hex: 0x%hx\n",ch);

}

**Q26.Read a small int into char variable and print this small int as character.**

#include<stdio.h>

int main()

{

char ch;

printf("enter the small integer\n");

scanf("%c", &ch);

printf("%hhd\n",ch);

}

**Q27.  Read a character into char variable and print the previous character     and the following character.**

#include<stdio.h>

int main()

{

char ch;

printf("enter the character\n");

scanf("%c",&ch);

printf("previous *character* is %c\n", ch-1);

printf("following character is %c\n", ch+1);

}

**Q28.Ask user to enter a capital letter. Read the capital letter into a char    variable. Set the 5th bit in this char variable to 1. Now print this char     as character**

#include<stdio.h>

int main()

{

char ch;

printf("Kinldy enter the capital letter\n");

scanf("%c", &ch);

ch |=(1<<4);

printf("modified character after seted the 5th bit is: %c\n",ch);

}

**Q29.Ask user to enter a small letter. Read the small letter into a char     variable. Clear the 5th bit in this char variable to 0. Now print this char     variable as character**

#include<stdio.h>

int main()

{

char ch;

printf("Kinldy enter the small letter\n");

scanf("%c", &ch);

ch &=~(1<<4);

printf("modified character after clearing the 5th bit is: %c\n",ch);

}

**Q30. Read a short variable from the user as hex number. Print the value of**

**bit number 0 and bit number 9. Note that value of bit will be either zero**

**or one.\*/**

#include<stdio.h>

int main()

{

short int num, bit\_0, bit\_9;

printf("enter the Hexa number:\n");

scanf("%hx", &num);

bit\_0= num&1;

bit\_9= (num>>9)&1;

printf("Value at bit 0: %hd\n", bit\_0);

printf("Value at bit 9: %hd\n", bit\_9);

}

**Q30. Read a short variable from the user as hex number. Clear the 5th bit to 0**

**and set the 12th bit to 1. After clearing and setting the bits, print the**

**value as hex number. Verify that the 5th bit is cleared to zero and 12th**

**bit is set to 1. \*/**

#include<stdio.h>

int main()

{

short int num, bit\_5, bit\_12;

printf("enter the Hexa number:\n");

scanf("%hx", &num);

num &= ~(1<<4);

num |= (1<<11);

printf("modifies value : %hx\n", num);

bit\_5= (num>>4) &1;

bit\_12= (num>>11)&1;

//if(bit\_5==0)

printf("5th bit cleared:%hd\n",bit\_5==0);

//if(bit\_12==1)

printf("12th bit set:%hd\n", bit\_12==1);

}

**Q31. Read a short variable from the user as hex number. Clear the 5th bit to 0**

**and set the 12th bit to 1. After clearing and setting the bits, print the**

**value as hex number. Verify that the 5th bit is cleared to zero and 12th**

**bit is set to 1. \*/**

#include<stdio.h>

int main()

{

short int num, bit\_5, bit\_12;

printf("enter the Hexa number:\n");

scanf("%hx", &num);

num &= ~(1<<4);

num |= (1<<11);

printf("modifies value : %hx\n", num);

bit\_5= (num>>4) &1;

bit\_12= (num>>11)&1;

//if(bit\_5==0)

printf("5th bit cleared:%hd\n",bit\_5==0);

//if(bit\_12==1)

printf("12th bit set:%hd\n", bit\_12==1);

}

**Q32. Read a short variable from the user as hex number. Also read the bit**

**position to set, from the user. Now set the specified bit in the hex number.**

**Finally print the number in hex, after setting the given bit. \*/**

#include<stdio.h>

int main()

{

short int num;

int bitpos;

printf("enter the Hexa number:\n");

scanf("%hx", &num);

printf("enter the bitpos:\n");

scanf("%d", &bitpos);

num |= (1<<bitpos);

printf("Hex value after setting the bit is :ox%hx\n",num);

}

Q33. Read a short variable from the user as hex number. Also read the bit

position to clear, from the user. Now clear the specified bit in the hex

number. Finally print the number in hex, after clearing the given bit. \*/

#include<stdio.h>

int main()

{

short int num;

int bitpos;

printf("enter the Hexa number:\n");

scanf("%hx", &num);

printf("enter the bitpos:\n");

scanf("%d", &bitpos);

num &= ~(1<<bitpos);

printf("Hex value after setting the bit is :ox%hx\n",num);

}

**Q34.Read a short variable from the user as hex number. Also read the**

**bit position, from the user. Print the value present in the 4 bits,**

**starting from the given bit position. \*/**

#include<stdio.h>

int main()

{

short int num;

int bitpos,i;

printf("enter the Hexa number:\n");

scanf("%hx", &num);

printf("enter the bitpos:\n");

scanf("%d", &bitpos);

for (i=0;i<=16;i++)

printf("%d",(num>>i)&1);

bitpos=num>>4&1;

printf("the value present in the 4h bit is %d\n",bitpos);

}

**Q35. An electric pole is present on one side of the road. From the opposite**

**side of the road, the tip of the pole is making certain angle. Ask the**

**user to enter the angle in degrees and width of the road in meters.**

**Calculate the height of the pole and print it.**

**Formula: TAN theta = pole\_height / road\_width**

**Use math function tan().\*/**

#include<stdio.h>

#include<math.h>

int main()

{

double angle, width, height;

printf("enter the angle in degree:\n");

scanf("%lf", &angle);

printf("Enter the width of the road in meters:\n");

scanf("%lf", &width);

height= width\*tan(angle);

printf("the height of the pole is: %lf\n", height);

}

**Q36. Read the time from the user as hours, minutes and seconds. Convert**

**the time to total number of seconds.\*/**

#include<stdio.h>

int main()

{

float hour, minute, sec;

int total\_sec;

printf("Enter the hour:\n");

scanf("%f", &hour);

printf("Enter the minute:\n");

scanf("%f", &minute);

printf("Enter the Sconds:\n");

scanf("%f", &sec);

total\_sec= (hour\*3600)+(minute\*60)+sec;

printf("total\_sec: %d\n", total\_sec);

}

**Q37.Read the total number of seconds from the user. And display it as time**

**like hh:mm:ss.\*/**

#include<stdio.h>

int main()

{

int hr, min, sec;

printf("Enter the total number of sec:\n");

scanf("%d", &sec);

hr=sec/3600;

sec=sec%3600;

min=sec/60;

sec=sec%60;

printf("HH:MM:SS : %02d:%02d:%02d\n", hr, sec,min);

}

**ASSIGMENT 2. Assignments on If and Switch statements**

**Q1.Read an integer number from the user and print whether the number is**

**even number or odd number. Check the first bit (0th bit), if first bit**

**is one, the number is odd, else number is even\*/**

#include<stdio.h>

int main()

{

int num;

printf("Enter the number\n");

scanf("%d", &num);

if (num%2==0)

{

printf("%d is a even number\n",num);

}

else

{

printf("%d is a odd number\n",num);

}

}

**A2 Q2. Read an integer number from the user and print whether the number is**

**even number or odd number. Find the remainder value when the given**

**number is divided by 2. If remainder is zero, the number is even.**

**Else number is odd**

#include<stdio.h>

int main()

{

int num, rem=0;

printf("Enter the number\n");

scanf("%d", &num);

rem=num%2;

if (rem==0)

{

printf("%d is a even number\n",num);

}

else

{

printf("%d is a odd number\n",num);

}

}

**A2 Q3. Read an integer number from the user and print whether the number is**

**negative or positive. Find the number as negative or positive by checking**

**the most significant bit. That is bit 31. If bit 31 is one, it is**

**negative number. Else positive number.\*/**

#include<stdio.h>

int main()

{

int num;

printf("enter the number\n");

scanf("%d", &num);

if((num>>31)&1==1)

{

printf("%d is a negative number\n", num);

}

else

{

printf("%d is a postive number\n", num);

}

}

**A2 Q4 Read a character from the user, and print whether the character is**

**Capital letter / Small letter / Numeric digit / Other\*/**

#include<stdio.h>

int main()

{

char ch;

printf("enter the character:\n");

scanf("%c", &ch);

if (ch>=65 && ch<=90)

printf("%c is a Capital letter\n",ch);

else if(ch>=97 && ch<=122)

printf("%c is a small letter\n", ch);

else if(ch>=48 && ch<=57)

printf("%c is a numeric digit\n",ch);

else

printf("%c is a special character\n",ch);

}

**A2 Q5. Read a character from the user. If character is capital letter, convert**

**it to small letter and print it. If character is small letter, convert it**

**to capital letter. If character is nither capital nor small letter, print**

**as it is\*/**

#include<stdio.h>

int main()

{

char ch;

printf("enter the character:\n");

scanf("%c", &ch);

if(ch>=65 && ch<=90)

{

ch ^= 32;

printf("converted to lower case:%c\n",ch);

}

else if(ch>=97 && ch<=122)

{

ch ^=32;

printf("converted to upper case:%c\n", ch);

}

else

printf("No conversion needed: %c\n",ch);

}

**A2 Q6. Write a program to perform divisibility test. Read two numbers from the**

**user. Test if first number is divisible by second number without any**

**remainder. Based on the resule, print "Divisible Test Passed" or**

**"Divisible test failed".\*/**

#include<stdio.h>

int main()

{

int num1, num2;

printf("enter the two numbers:\n");

scanf("%d %d", &num1, &num2);

if(num1%num2==0)

{

printf("Divisible test Passed\n");

}

else

{

printf("Divisible test failed\n");

}

}

**A2 Q7.Read two integer numbers from the user. Print the bigger number. If both**

**numbers are equal, print "Both numbers are equal\n"**

#include<stdio.h>

int main()

{

int num1, num2;

printf("enter the Ist integer number:\n");

scanf("%d", &num1);

printf("enter the 2nd integer number:\n");

scanf("%d", &num2);

if(num1==num2)

{

printf("%d is equal to %d\n",num1,num2);

}

else if(num1>num2)

{

printf("%d is bigger then %d\n", num1,num2);

}

else

{

printf("%d is smaller the %d\n", num1,num2);

}

}

**A2 Q8. three integer numbers from the user. If these three numbers are**

**in asending order, print "Asending". If the numbers are in decending order**

**print "Decending". If all three numbers are equal print "Equal". If none**

**of the above are true, print "Random".\*/**

#include<stdio.h>

int main()

{

int num1, num2, num3;

printf("enter the 1st number:\n");

scanf("%d", &num1);

printf("enter the 2nd number:\n");

scanf("%d", &num2);

printf("enter the 3rd number:\n");

scanf("%d", &num3);

if(num1==num2 && num2==num3)

{

printf("Equal\n");

}

else if(num1<num2 && num2<num3)

{

printf("Ascending\n");

}

else if(num1>num2 && num2>num3)

{

printf("Decending\n");

}

else

{

printf("Random\n");

}

}

**A2 Q9. Read the radius of a sphere in centimeters from the user. Ask the user**

**to enter the metal type of sphere as below:**

**printf("1:Aluminum 2:Copper 3:Gold 4:Iron 5:Silver\n");**

**Read the metal type as number 1 to 5. If user enters invalid metal number**

**print error and exit from the program.**

**Calculate and print the weight of sphere by using following densities of**

**metals:**

**Aluminum 2.712 grams/cm3 (grams per centimeter cube)**

**Copper 8.940**

**Gold 19.320**

**Iron 7.850**

**Silver 10.490\*/**

#include<stdio.h>

#define PI 3.14

int main()

{

double rad,weight;

int metal;

printf("Enter the radius of a sphere in cm\n");

scanf("%lf", &rad);

printf("1:Aluminum 2:Copper 3:Gold 4:Iron 5:Silver\n");

printf("Enter the metal\n");

scanf("%d", &metal);

switch(metal)

{

case 1: weight=2.712\*(4.0/3.0)\*PI\*rad\*rad\*rad;

break;

case 2: weight=8.940\*(4.0/3.0)\*PI\*rad\*rad\*rad;

break;

case 3: weight=19.320\*(4.0/3.0)\*PI\*rad\*rad\*rad;

break;

case 4: weight=7.850\*(4.0/3.0)\*PI\*rad\*rad\*rad;

break;

case 5: weight=10.490\*(4.0/3.0)\*PI\*rad\*rad\*rad;

break;

default:printf("invalid input\n");

}

printf("the weight of sphere: %lf\n", weight);

}

**A2 Q10. Read two integer numbers from the user. Ask the user to enter any**

**arithmetic operator character. User can enter any one of the following**

**arithmetic operator character:**

**+ ­ \* / %**

**Based on the character do the corresponding operation on the numbers and**

**print the result. Use If­elseIf­elseIf statement.\*/**

#include<stdio.h>

int main()

{

int num1, num2,result;

char ch;

printf("Enter the 1st integer num:\n");

scanf("%d", &num1);

printf("Enter the 2nd integer num:\n");

scanf("%d", &num2);

printf("Enter any arithmetic operator\n");

printf("1. +\n");

printf("2. -\n");

printf("3. \*\n");

printf("4. /\n");

printf("5. %%\n");

scanf(" %c", &ch);

if(ch=='+')

result=num1+num2;

else if(ch=='-')

result=num1-num2;

else if(ch=='\*')

result=num1\*num2;

else if(ch=='/')

result=num1/num2;

else if(ch=='%')

result=num1%num2;

else

printf("invalid input\n");

printf("result=%d\n",result);

}

**A2 Q10. Read two integer numbers from the user. Ask the user to enter any**

**arithmetic operator character. User can enter any one of the following**

**arithmetic operator character:**

**+ ­ \* / %**

**Based on the character do the corresponding operation on the numbers and**

**print the result. Use If­elseIf­elseIf statement.\*/**

#include<stdio.h>

int main()

{

int num1, num2,result;

char ch;

printf("Enter the 1st integer num:\n");

scanf("%d", &num1);

printf("Enter the 2nd integer num:\n");

scanf("%d", &num2);

printf("Enter any arithmetic operator\n");

printf("1. +\n");

printf("2. -\n");

printf("3. \*\n");

printf("4. /\n");

printf("5. %%\n");

scanf(" %c", &ch);

if(ch=='+')

result=num1+num2;

else if(ch=='-')

result=num1-num2;

else if(ch=='\*')

result=num1\*num2;

else if(ch=='/')

result=num1/num2;

else if(ch=='%')

result=num1%num2;

else

printf("invalid input\n");

printf("result=%d\n",result);

}

**A2 Q11. Read two integer numbers from the user. Ask the user to enter any**

**arithmetic operator character. User can enter any one of the following**

**arithmetic operator character:**

**+ ­ \* / %**

**Based on the character do the corresponding operation on the numbers and**

**print the result. Use switch statement.\*/**

#include<stdio.h>

int main()

{

int num1, num2,result;

char ch;

printf("Enter the 1st integer num:\n");

scanf("%d", &num1);

printf("Enter the 2nd integer num:\n");

scanf("%d", &num2);

printf("Enter any arithmetic operator\n");

printf("1. +\n");

printf("2. -\n");

printf("3. \*\n");

printf("4. /\n");

printf("5. %%\n");

scanf(" %c", &ch);

switch(ch)

{

case '+':result=num1+num2;

printf("%d + %d= %d\n", num1,num2,result);

break;

case '-':result=num1-num2;

printf("%d - %d= %d\n", num1,num2,result);

break;

case '\*':result=num1\*num2;

printf("%d \* %d= %d\n", num1,num2,result);

break;

case '/':result=num1/num2;

printf("%d / %d= %d\n", num1,num2,result);

break;

case '%':result=num1%num2;

printf("%d %% %d= %d\n", num1,num2,result);

break;

default:

printf("invalid input\n");

}

}

**A2 Q12.c Read month from the user as integer. User should enter 1 for the month**

**January, 2 for February and so on. The program should print the number**

**days in the given month. Use switch() statement. Assume that year is**

**not a leaf year.\*/**

#include<stdio.h>

int main()

{

int month;

printf("Enter the month:\n");

printf("1:Jan 2:Feb 3:Mar 4:Apr 5:May 6:jun 7:july 8:Aug 9:Sep 10:Oct 11:Nov 12:Dec\n ");

scanf("%d", &month);

switch(month)

{

case 1:

printf("January has 31 days.\n");

break;

case 2:

printf("February has 28 days.\n");

break;

case 3:

printf("March has 31 days.\n");

break;

case 4:

printf("April has 30 days.\n");

break;

case 5:

printf("May has 31 days.\n");

break;

case 6:

printf("june 30 days.\n");

break;

case 7:

printf("July has 31 days.\n");

break;

case 8:

printf("August has 31 days.\n");

break;

case 9:

printf("September has 30 days.\n");

break;

case 10:

printf("october has 31 days.\n");

break;

case 11:

printf("November has 30 days.\n");

break;

case 12:

printf("December has 31 days.\n");

break;

default:

printf("Error invalid month\n");

}

}

**A2 Q13.c Read month from the user as integer. User should enter 1 for the month**

**January, 2 for February and so on. The program should print the number**

**days in the given month. Use switch() statement. Assume that year is**

**not a leaf year.**

**Q13. Redo the above program. Now read both month and year from the user.**

**Print the number of days in the given month. Assume that year is leaf**

**year if remainder is zero when devided by 4.\*/**

#include<stdio.h>

int main()

{

int month, years;

printf("Enter the month:\n");

printf("1:Jan 2:Feb 3:Mar 4:Apr 5:May 6:jun 7:july 8:Aug 9:Sep 10:Oct 11:Nov 12:Dec\n ");

scanf("%d", &month);

printf("Enter the year:\n");

scanf("%d", &years);

switch(month)

{

case 2: if((years%4==0 && years%100!=0) ||(years%400==0))

printf("The number of day in the given month is 29 Days\n");

else

printf("The number of day in the given month is 28 Days\n");

break;

case 1:

printf("January has 31 days.\n");

break;

case 3:

printf("March has 31 days.\n");

break;

case 4:

printf("April has 30 days.\n");

break;

case 5:

printf("May has 31 days.\n");

break;

case 6:

printf("june 30 days.\n");

break;

case 7:

printf("July has 31 days.\n");

break;

case 8:

printf("August has 31 days.\n");

break;

case 9:

printf("September has 30 days.\n");

break;

case 10:

printf("october has 31 days.\n");

break;

case 11:

printf("November has 30 days.\n");

break;

case 12:

printf("December has 31 days.\n");

break;

default:

printf("Error invalid month\n");

}

}

**3. Assignments on Arrays and Loops**

**A3 Q1.Define an array of 5 int variables. Ask user to enter 5 numbers.**

**Read five numbers into this array of five. Finally print these 5 numbers.**

#include<stdio.h>

int main()

{

int arr[5],i;

printf("Enter Five numbers:\n");

for(i=0;i<5;i++)

scanf("%d", &arr[i]);

printf("the numbers you entered are\n");

for(i=0;i<5;i++)

printf("%d\n",arr[i]);

}

/\***A3 Q2. Read five numbers from the user into an array of 5. Calculate the smallest**

**number, largest number, total and average. Finally print these calculated**

**values.\*/**

#include<stdio.h>

int main()

{

int arr[5],total,small,large;

float avg;

int i;

printf("enter the five numbers:\n");

for(i=0;i<5;i++)

scanf("%d", &arr[i]);

small =arr[0];

large =arr[0];

total=0;

for(i=0;i<5;i++)

{

if(arr[i]>large)

large=arr[i];

if(arr[i]<small)

small=arr[i];

}

printf("smallest number is: %d\n",small);

printf("largest number is: %d\n",large);

for(i=0;i<5;i++)

{

total+=arr[i];

}

avg= total/5;

printf("total:=%d\n", total);

printf("Average:=%.2f\n", avg);

}

/\* **A3 Q3 Define an array of 6 char variables. Ask the user to enter the marks of**

**6 subjects. Read these marks into the array. As each mark is from 0 to 100,**

**it can be stored in 8 bit integer, that is char variable.**

**Print whether the student is passed or failed. Take pass mark as 35\*/**

#include<stdio.h>

int main()

{

char arr[6];

int i;

int pass\_marks=35;

int ispassed=1;

printf("enter the marks of the 6 sub which is from 0 to 100:\n");

for(i=0;i<6;i++)

{

scanf("%hhd", &arr[i]);

}

for(i=0;i<6;i++)

{

if(arr[i]<pass\_marks)

{

ispassed=0;

break;

}

}

if(ispassed)

{

printf("the student has passed\n");

}

else

{

printf("the student has failed\n");

}

return 0;

}

**/\*A3 Q4. Read a a line of string from the user. Find the length of the string and**

**print.\*/**

#include<stdio.h>

#include<string.h>

int main()

{

char str[80];

printf("Enter the string\n");

fgets(str,80,stdin);

int len=strlen(str);

printf("the length of the string is:%d\n", len);

}

**/\* A3 Q5. Read a line from the user and print the number of capital letters present**

**in the string.\*/**

#include<stdio.h>

#include<string.h>

int main()

{

char str[80];

int i, cnt=0;

printf("Enter the string\n");

fgets(str, 80, stdin);

for(i=0;str[i];i++)

{

if(str[i]>=65 && str[i]<=90)

cnt++;

}

printf("the number of capital letters present in the string is:%d\n",cnt);

}

**/\* A3 Q6. Read a line from the user and print the number of spaces present in the line.\*/**

#include<stdio.h>

#include<string.h>

int main()

{

char str[100];

int cnt=0, i;

printf("Enter the string\n");

fgets(str, 100, stdin);

for(i=0;str[i];i++)

{

if(str[i]==32)

cnt++;

}

printf("the number of spaces present in the line is:%d\n",cnt);

}

**/\*A3 Q7. Read a line and print the number of words present in the line string.**

**Note that one or more spaces could be present between the wokds.\*/**

#include<stdio.h>

#include<string.h>

int main()

{

char str[100];

int cnt=1, i;

printf("enter the string\n");

fgets(str, 100, stdin);

for(i=0;str[i];i++)

{

if(str[i]==' ' && str[i+1]==' ')

memmove(str+i, str+i+1,strlen(str+i+1)+1);

}

for(i=0;str[i];i++)

{

if (str[i]==' ' || str[i] =='\0')

cnt++;

}

printf("the number of words present in the line string is %d\n",cnt);

}

**/\* A3 Q3 Read a line from the user, and convert each capital letter to small letter and small letter to capital. All the other characters should remain as**

**it is. Finally print the modified line string.\*/**

#include<stdio.h>

#include<string.h>

int main()

{

char str[100];

int i;

printf("enter the string:\n");

scanf("%[^\n]",str);

for(i=0;str[i];i++)

{

if ((str[i]>=65 && str[i]<=90) || (str[i]>=97 && str[i]<=122))

str[i]^=' ';

}

puts(str);

}

**/\*A3 Q9 Define two strings, each of size 80 chars. Read a line string into**

**first string. Copy this first string into the second one. Print the**

**second string.\*/**

#include<stdio.h>

#include<string.h>

int main()

{

char str1[80];

char str2[80];

puts("enter the first string\n");

fgets(str1,80,stdin);

puts("enter the second string\n");

fgets(str2,80,stdin);

strcpy(str2,str1);

puts(str2);

}

**/\*A3 Q10. Define a string of size 80 chars. Also define 5 strings of size 20 chars**

**each. Ask the user to enter a line containg 5 words. Copy each word**

**into individual strings. Finally print all the inidividual string.\*/**

#include<stdio.h>

#include<string.h>

int main()

{

char str[80],str1[20],str2[20],str3[20],str4[20],str5[20];

puts("Enter a line containg 5 words:");

fgets(str,80,stdin);

int i,j;

for(i=0;str[i];i++)

{

if(str[i]==' ')

{

i++;

break;

}

str1[i]=str[i];

}

str1[i]='\0';

for(j=0;str[i];i++,j++)

{

if(str[i]==' ')

{

i++;

break;

}

str2[j]=str[i];

}

str2[j]='\0';

for(j=0;str[i];i++,j++)

{

if(str[i]==' ')

{

i++;

break;

}

str3[j]=str[i];

}

str3[j]='\0';

for(j=0;str[i];i++,j++)

{

if(str[i]==' ')

{

i++;

break;

}

str4[j]=str[i];

}

str4[j]='\0';

for(j=0;str[i];i++,j++)

{

if(str[i]==' ')

{

i++;

break;

}

str5[j]=str[i];

}

str5[j]='\0';

puts(str1);

puts(str2);

puts(str3);

puts(str4);

puts(str5);

}

**4. Assignments on defining and using structure variables**

**/\*A4 Q1. Define a structure variable of type 'struct Time' as given below:**

**struct Time**

**{**

**char hour;**

**char min;**

**char sec;**

**};**

**Note that the structure fields hour, min and sec are char (small) integers.**

**Small integers are sufficient to store the time values.**

**Ask the user to enter hours, minutes and seconds. Read those values into**

**the structure. Finally print the time structure with the format 'HH:MM:SS'.\*/**

#include<stdio.h>

struct Time

{

char hour;

char min;

char sec;

};

int main()

{

struct Time time;

{

printf("enter hour:\n");

scanf("%hhd", &time.hour);

printf("enter Min:\n");

scanf("%hhd", &time.min);

printf("enter sec\n");

scanf("%hhd", &time.sec);

printf("Time structure : %.2hhd: %.2hhd: %.2hhd\n", time.hour, time.min, time.sec);

}

}

**/\* A4 Q2. Define a structure variable of type 'struct Date' as given below:**

**struct Date**

**{**

**char date;**

**char month;**

**short year;**

**};**

**Ask the user to enter date and read into the structure variable.**

**Finally print the values in the structure variable in the format**

**DD/MM/YYYY'. Again print the date in the format 'DD MMM YYY', where**

**MMM is the name of the month in three characters like 'Jan' .. 'Dec'.\*/**

#include<stdio.h>

struct Date

{

char date;

char month;

short year;

};

char \*m[]={"Jan","Feb","March","Apr","May","Jun","Jul","Aug","Sep","Oct","Nov","Dec"};

int main()

{

struct Date dt;

printf("Enter the date: ");

scanf("%hhd", &dt.date);

printf("Enter the month: ");

scanf("%hhd", &dt.month);

printf("Enter the Year: ");

scanf("%hd", &dt.year);

printf("%.2d/ %.2d/%.2d\n", dt.date,dt.month,dt.year);

printf("%02d %s %02d\n",dt.date,m[dt.month-1],dt.year);

}

**/\* QA4 Q3. Define a structure variable of type 'struct Student' as given below:**

**struct Student**

**{**

**char name[40];**

**long long mobile;**

**struct Date dob; //Date of birth, struct Date as given in assignment**

**};**

**Ask the user to enter the student information, and fill the defined**

**structure variable. Finally print the studnent informaiton in the**

**following format:**

**Name : Bharat Kumar B**

**Mobile : 9393916923**

**DOB : 18/10/19**

**Note that, name may have multiple words.\*/**

#include<stdio.h>

struct Date

{

int day;

int month;

int year;

};

struct Student

{

char name[40];

long long mobile;

struct Date dob;

};

int main()

{

struct Student student;

printf("enter the student information\n");

printf("Name : ");

fgets(student.name, sizeof(student.name),stdin);

printf("Mobile : ");

scanf("%lld", &student.mobile);

printf("DOB : ");

scanf("%d/%d/%d", &student.dob.day,&student.dob.month, &student.dob.year);

printf("Student information\n");

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

printf("Mobile : %lld\n", student.mobile);

printf("DOB : %02d/%02d/%02d\n", student.dob.day, student.dob.month,student.dob.year );

}

**A4Q4. Define an array of 5 structure variables of type 'struct Student'    Read from the user, information of 5 students in to this array.    Finally print the array in the following format:**

**S.No    Name              Mobile      Date of Birth    ­­­­    ­­­­­­­­­­­­­­­­­ ­­­­­­­­­­  ­­­­­­­­­­­­­       1    Bharat Kumar B    9393916923  18/10/1994       ­­    ­­                ­­         ­­**

**5    xxxxx xxxxxx x    xxxxxxxxxx  xx/xx/xxxx**

#include<stdio.h>

struct Student

{

int Srno;

char name[40];

long long mobile;

char dob[11];

};

int main()

{

struct Student st[5];

int i;

for(i=0;i<5;i++)

{

printf("Enter details:\n");

printf("Enter Sr.No:\n");

scanf("%d",&st[i].Srno);

printf("Enter the Student Name:\n");

scanf(" %[^\n]s",st[i].name);

printf("Enter Mobile number:\n");

scanf("%lld",&st[i].mobile);

printf("Enter the DOB DD/MM/YYYY:\n");

scanf("%s",st[i].dob);

}

printf("S.No Name Mobile Date of Birth\n");

printf("--------------------------------------------------------\n");

for(i=0;i<5;i++)

{

printf("%d %s %lld %s\n",st[i].Srno,st[i].name,st[i].mobile,st[i].dob);

}

}

**5. Assignments on defining simple functions**

**/\*A5 Q1.Write the following function:**

**double getSphereVolume(double radius);**

**This function takes the radius of sphere as parameter and returns the**

**volume of sphere. Write a main() function to test this function. To test**

**this function, in main() function read radius from the user and call this**

**function. Finally print the value returned by this function.**

**Note that, inside the getSphereVolume() function, you should not print the**

**volume. This function should calculate and return the volume. Printing of**

**volume should be done only from the main() function.\*/**

#include<stdio.h>

#define PI 3.14

double getSphereVolume(double radius)

{

return (4.0/3.0) \* PI \* radius \* radius \* radius;

}

int main()

{

double radius, volume;

printf("enter the radius of the sphere\n");

scanf("%lf", &radius);

volume= getSphereVolume(radius);

printf("volume of sphere is : %lf\n", volume);

}

**/\*A5 Q2\*Write the following function:**

**double getSphereSurfaceArea(double radius);**

**Write the main() function to call and test the above function\*/**

#include<stdio.h>

#define PI 3.14

double getSphereSurfaceArea(double radius)

{

return 4.0 \* PI \* radius \* radius;

}

int main()

{

double radius, volume;

printf("enter the radius of the sphere\n");

scanf("%lf",&radius);

volume=getSphereSurfaceArea(radius);

printf("Area of sphere id :%lf\n",volume);

}

**/\*A5 Q3. Write the following function:**

**double getSphereWeight(double radius, double density);**

**Write the main() function to call and test the above function.\*/**

#include<stdio.h>

#define PI 3.14

double getSphereWeight(double radius, double density)

{

double volume;

volume= (4.0/3.0) \* PI \* radius \* radius \* radius;

return density \* volume;

}

int main()

{

double radius, density, weight;

printf("Enter the radius of the sphere:\n");

scanf("%lf", &radius);

printf("Enter the density:\n");

scanf("%lf", &density);

weight = getSphereWeight(radius,density);

printf("Weight of sphere is:%lf\n",weight);

}

**/\* A5 Q4. wing function. This function sets the bit at the given**

**position to one.**

**unsigned int setBit(unsigned int val , int bitPos);**

**Write the main() function to call and test the above function.**

**In the main() function read an hexadecimal number from the user and also**

**read bit position from the user. Call the above function. Print the valu**

**returned by that function in hexadecimal.**

**Note that reading hex number and printing hex number allows us to check if**

**the bit is set correctly or not.\*/**

#include<stdio.h>

unsigned int setBit(unsigned int val , int bitPos)

{

val|=1<<bitPos;

return val;

}

int main()

{

int value;

int bitpostion;

printf("enter the hexa value:\n");

scanf("%x", &value);

printf("enter the bitpos\n");

scanf("%d", &bitpostion);

printf("the bit is set correctly after: %x\n", (value,bitpostion));

}

**/\*A5 Q5. Write the following function. This function clears the bit at the given**

**position to zero.**

**unsigned int clearBit(unsigned int val , int bitPos);**

**Write the main() function to call and test the above function.**

**In the main() function read an hexadecimal number from the user and also**

**read bit position from the user. Call the above function. Print the value**

**returned by that function in hex\*/**

#include<stdio.h>

unsigned int clearBit(unsigned int val , int bitPos)

{

val &=~(1<<bitPos);

return val;

}

int main()

{

int value, bitpostion,res;

printf("enter the hexa value:\n");

scanf("%x", &value);

printf("enter the bitpos:\n");

scanf("%d", &bitpostion);

res= clearBit(value,bitpostion);

printf("the bit is cleared after:%x\n",res);

}

**/\*A5 Q6. Write the following function. This function returns total number of seconds**

**elapsed at the given time**

**int getTotalSecondsElapsed(short hour, short min, short sec);**

**Write the main() function to call and test the above function.\*/**

#include<stdio.h>

int getTotalSecondsElapsed(short hour, short min, short sec)

{

return (hour\*3600+min\*60+sec);

}

int main()

{

short hr,minute,second;

int res;

printf("Enter the hour:\n");

scanf("%hd", &hr);

printf("Enter the minute:\n");

scanf("%hd", &minute);

printf("Enter the seconds:\n");

scanf("%hd", &second);

res= getTotalSecondsElapsed(hr,minute,second);

printf("Total seconds:%d\n", res);

}

**/\*A5 Q7. Write the following function. This function is same as function written in**

**the above assignment. But takes structure Time as parameter.**

**Struct Time is specified in the assignments on structures.**

**int getTotalSecondsElapsed(struct Time t);**

**Write the main() function to call and test the above function.\*/**

#include<stdio.h>

struct Time{

short hour;

short min;

short sec;

};

int getTotalSecondsElapsed(struct Time t)

{

return (t.hour\*3600+t.min\*60+t.sec);

}

int main()

{

struct Time t;

printf("enter the hour:\n");

scanf("%hd", &t.hour);

printf("enter the minute:\n");

scanf("%hd", &t.min);

printf("enter the seconds:\n");

scanf("%hd", &t.sec);

int seconds=getTotalSecondsElapsed(t);

printf("Total seconds:%d\n", seconds);

}

**/\*A5 Q8.Write the following function. This function is reverse of previous**

**function. It takes total seconds as input and returns struct time.**

**struct Time getTotalSecondsElapsed(int totalSeconds);**

**Write the main() function to call and test the above function.\*/**

#include<stdio.h>

struct Time{

short hour;

short min;

short sec;

};

struct Time getTotalSecondsElapsed(int totalSeconds)

{

struct Time time;

time.hour=totalSeconds/3600;

time.min=(totalSeconds%3600)/60;

time.sec=(totalSeconds%3600)%60;

return time;

}

int main()

{

struct Time t;

int totalseconds;

printf("Enter the total seconds\n");

scanf("%d", &totalseconds);

t=getTotalSecondsElapsed(totalseconds);

printf("Time: %02d:%02d:%02d\n", t.hour,t.min,t.sec);

}

**6. Assignments on using pointer variables**

**/\*A6 Q1. Define an integer pointer variable with name 'pi'. Define this pointer**

**as global variable, not local variable. Now print the following:**

**­ Size of pointer variable using sizeof operator**

**­ Print the address of this pointer variable**

**­Print the value present in this pointer variable (which is supposed to be an address).**

**Could you guess, what will be the value, present in this pointer variable ?\*/**

#include<stdio.h>

int \*pi;

int main()

{

printf("Size of the pointer variable is:%lu\n", sizeof(pi));

printf("Address of the pointer variable is:%p\n", &pi);

printf("Value of the pointer variable is:%p\n",(void\*)pi);

}

**/\*A6 Q2.Define an integer pointer variable. Assign any one number into this**

**pointer variable. For example put 100 into this pointer variable.**

**You need to typecast 100 as a pointer to avoid the warning.**

**Next print the value in the pointer variable. It should print the value**

**you have assigned. Now store some other number into this pointer and**

**again print it.\*/**

#include<stdio.h>

int main()

{

int \*ptr;

ptr=(int\*)100;

printf("value at the pointer variable is :%d\n", \*ptr);

\*ptr=(int\*)200;

printf("New value at the pointer variable is :%d\n", \*ptr);

}

**/\*A6 Q3. Define three pointer pointer variables pi, ps and pc. One is integer**

**pointer, second is short pointer and last is char pointer. Print the**

**size of each pointer using sizeof operator. What do you expect the size of these pointers?**

**Now assign all these three pointers with a value of 100. Use typecast**

**to avoid warning. Now print all the three pointers. Next increment or**

**add one (pi++; or pi = pi + 1;) to all three pointer variables.**

**After incrementing, print the values of all three pointers. You should**

**be able to expect the values it is going to print.\*/**

#include<stdio.h>

int main()

{

int \*pi;

short \*ps;

char \*pc;

printf("Size of integer pointer: %lu bytes\n", sizeof(pi));

printf("Size of short pointer: %lu bytes\n", sizeof(ps));

printf("Size of char pointer: %lu bytes\n", sizeof(pc));

pi=(int\*)100;

ps=(short\*)100;

pc=(char\*)100;

printf("Integer pointer: %p\n", pi);

printf("Short pointer: %p\n", ps);

printf("Char pointer: %p\n", pc);

pi++;

ps++;

pc++;

printf("After incrementing:\n");

printf("Integer pointer: %p\n", pi);

printf("Short pointer: %p\n", ps);

printf("Char pointer: %p\n", pc);

}

**/\* A6 Q4. Define an integer pointer variable. Ask user to enter some number or**

**address (Please note that address is also a number only). Now use**

**scanf() function and read the number into this pointer variable.**

**Finally print this pointer varible.**

**Note: You are always doing this with integer variables, that is reading**

**a number from the user. Now you are doing the same thing with pointer**

**variable. That is you are reading an address (number) from the user**

**into this pointer variable.\*/**

#include<stdio.h>

int main()

{

int \*ptr,num;

printf("enter the number\n");

scanf("%d",&num);

ptr=(int\*)num;

printf("value at pointer variable is :%p\n",ptr);

}

**/\*A6 Q5.Define one integer variable and one pointer to integer variable.**

**Store the address of integer variable into pointer variable.**

**­ Assign a value of 100 to integer variable.**

**­ Print the content of integer variable, using pointer variable.**

**­ Assign a new value (ex: 200) into integer variable by using pointer.**

**­ Print the new value in integer, using integer variable.**

**­ Read a new value into integer from user, using scanf() function**

**­ Print the new value in integer using pointer variable.**

**Read a new value into integer, through pointer variable using scanf() function**

**Print the new value in integer using integer variable \*/**

#include<stdio.h>

int main()

{

int num;

int \*ptr=&num;

\*ptr=100;

printf("content of integer variable is %d\n", num);

\*ptr=200;

printf("new value into integer is: %d\n", num);

printf("enter a new value:\n");

scanf("%d",&num);

printf("new value into integer using ppointer variable: %d\n", \*ptr);

printf("enter a new value for the integer using pointer variable:\n");

scanf("%d", ptr);

printf("New value in integer using integer variable: %d\n", num);

}

**/\*A6 Q6. Define an array of 5 integers and one integer pointer.**

**Store the starting address of array into pointer variable.**

**­ Using scanf() read 5 numbers into the array, from the user**

**­ Print these 5 numbers by using, pointer variable**

**­ Again read 5 numbers from the user, now through the pointer variable**

**­ Print the values in the array\*/**

#include<stdio.h>

int main()

{

int arr[5],i;

int \*ptr;

ptr=arr;

printf("enter the five numbers\n");

for(i=0;i<5;i++)

{

scanf("%d",&arr[i]);

}

for(i=0;i<5;i++)

{

printf("%d\n", ptr[i]);

}

printf("Enter array elements using pointer varible:\n");

for(i=0;i<5;i++)

{

scanf("%d",&ptr[i]);

}

for(i=0;i<5;i++)

{

printf("%d\n", arr[i]);

}

}

**/\*A6 Q7. Define a structure variable of type 'struct Student', and structure pointer**

**variable. Store the address of structure variable into the pointer variable.**

**struct Student**

**{**

**char name[40];**

**long long mobile;**

**int yearOfBirth; // Year in which one has born**

**// };**

**­ Using scanf() function read into the structure variable     ­ Print the content of the structure variable, using pointer variable     ­ Using scanf() function read into the structure, through pointer variable     ­ Print the content of structure variable using structure variable itself**

**\*/**

#include<stdio.h>

struct Student

{

char name[40];

long long mobile;

int yearOfBirth;

};

int main()

{

struct Student student;

struct Student \*ptr=&student;

printf("Enter the student name:\n");

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

printf("Enter mobile number:\n");

scanf("%lld", &student.mobile);

printf("Enter YearofBirth:\n");

scanf("%d",&student.yearOfBirth);

printf("Student Name : %s\n", ptr->name);

printf("Mobile number: %lld\n", ptr->mobile);

printf("Yearofbirth : %d\n",ptr->yearOfBirth);

printf("Enter student name using pointer variable: ");

scanf("%s", ptr->name);

printf("Enter student mobile number using pointer variable: ");

scanf("%lld", &ptr->mobile);

printf("Enter student year of birth using pointer variable: ");

scanf("%d", &ptr->yearOfBirth);

printf("Content of the structure using structure variable itself:\n");

printf("Student Name: %s\n", student.name);

printf("Student Mobile Number: %lld\n", student.mobile);

printf("Student Year of Birth: %d\n", student.yearOfBirth);

}

**/\*A6 Q7. Define an integer variable, one integer pointer variable, one short**

**pointer and one char pointer variables. Assign value of 0x12345678 to**

**the integer variable. Initialize all three pointers with the address of**

**the integer variable. Use type casting while assigning integer address to**

**short and char pointers.**

**Now using three pointer variables, print the values pointed by these**

**pointers. Expect the values that are going to print.**

**Increment the short pointer and print again**

**Increment the char pointer and print agai**

**Increment the char pointer and print again.**

**Increment the char pointer and print again.\*/**

#include<stdio.h>

int main()

{

int num= 0x12345678;

int \*intptr=&num;

short \*shortptr=(short\*)&num;

char \*charptr=(char \*)&num;

printf("value pointed by an integer pointer:0x%x\n", \*intptr);

printf("value pointed by an short pointer:0x%x\n", \*shortptr);

printf("value pointed by an char pointer:0x%x\n", \*charptr);

shortptr++;

printf("after incremented short pointer\n");

printf("value pointed by an short pointer:0x%x\n", \*shortptr);

charptr++;

printf("after incremented char pointer\n");

printf("value pointed by an char pointer:0x%x\n", \*charptr);

charptr++;

printf("after incremented char pointer again:\n");

printf("value pointed by an char pointer:0x%x\n", \*charptr);

charptr++;

printf("after incremented char pointer again:\n");

printf("value pointed by an char pointer:0x%x\n", \*charptr);

}

**/\*A6 Q9 Define an integer pointer variable. Using malloc() function, allocate**

**bytes of memory and store the allocated address in the pointer variable.**

**Now this pointer is pointing to the dynamically allocated integer.**

**Read a value from the user into this allocated memory using scanf().**

**Print the value present in the allocated memory.**

**Note: This assignment is similar to assignment 5. In assignment 5,**

**pointer is holding the address of integer variable. This integer variable**

**is statically allocated. In this assignment, pointer is pointing to**

**dynamically allocated variable. \*/**

#include<stdio.h>

#include<stdlib.h>

int main()

{

int num, \*ptr;

ptr=(int\*)malloc(sizeof(int));

if(ptr==NULL)

{

printf("Memory not allocated\n");

return 1;

}

printf("enter the value\n");

scanf("%d", &num);

ptr=&num;

printf("%d\n", \*ptr);

}

**/\*A6 Q10. Define an integer pointer variable. Using malloc() function allocate**

**20 bytes of memory (to hold 5 integers). Store the allocated address in**

**the pointer variable. So now this pointer varialbe is pointing to an array of 5 integers.**

**Read 5 integers from the user and fill the allocated array.**

**Print the values in the allocated array.**

**Note:This assignment is similar to assignment 6. In assignment 6, pointer**

**is pointing to statically allocated array. Here pointer is pointing**

**to dynamically allocated array.\*/**

#include<stdio.h>

#include<stdlib.h>

int main()

{

int \*ptr;

ptr=(int \*)malloc(5 \*sizeof(int));

if(ptr==NULL)

{

printf("memory not alloctaed\n");

return 0;

}

printf("enter 5 integer:\n");

for(int i=0; i<5;i++)

{

scanf("%d", &ptr[i]);

}

printf("values in the allocated array:\n");

for(int i=0;i<5;i++)

{

printf("%d\n",ptr[i]);

}

printf("\n");

free(ptr);

}

**/\*A6 Q11. Define a pointer to structure of type struct Student. Look at the**

**assignment 7 above, for 'struct Student' type. Using malloc() function**

**allocate memory for this structure variable and store the address in the pointer.**

**­ Read student information into this dynamically allocated structure**

**from the user**

**­ Print the content of dynamically allocated structure variable**

**Note:**

**This assignment is similar to assignment 7. There pointer is pointing**

**to the statically allocated structure variable. Statically allocated**

**variable is already existing and it will have its own name. And by**

**putting its address in pointer, we are making pointer to point to already**

**existing variable. So pointer is just using already existing variable.**

**In this way pointer is not very useful.**

**Now in this assignment, pointer is pointing to dynamically allocated**

**variable. This variable is allocated just now at run time. One can**

**access this dynamically allocated variable only through pointer.**

**Dynamically allocated variables do not have any name associated with**

**them. So these can be accessed only through the pointer variables.**

**This is the real use of pointer variables.\*/**

#include <stdio.h>

#include <stdlib.h>

struct Student {

char name[40];

long long mobile;

int yearOfBirth;

};

int main()

{

struct Student \*ptr;

ptr=(struct Student \*)malloc(sizeof(struct Student));

if(ptr==NULL)

{

printf("memory nor alloctaed\n");

return 0;

}

printf("enter the student name:\n");

scanf("%s", ptr->name);

printf("enter the mobile name:\n");

scanf("%lld", &ptr->mobile);

printf("enter the Year Of Birth name:\n");

scanf("%d", &ptr->yearOfBirth);

printf("student information:\n");

printf("Name : %s\n",ptr->name);

printf("Mobile : %lld\n",ptr->mobile);

printf("Year of Birth : %d\n",ptr->yearOfBirth);

free(ptr);

}

**/\* A6 Q12. Define a pointer to structure of type struct Student. Using malloc()**

**function allocate memory for 4 student structures. Store this address in**

**the poiner. Now pointer is pointing to an array of 4 student structures.**

**Now use this pointer, just like an array of 4 student structures.**

**­ Read 4 students information into this array of 4 structures from user.**

**­ Print the 4 students information entered by the user \*/**

#include<stdio.h>

#include<stdlib.h>

struct Student {

char name[40];

long long mobile;

int yearOfBirth;

};

int main()

{

struct Student \*ptr;

ptr=(struct Student \*)malloc(4 \*sizeof(struct Student));

if(ptr==NULL)

{

printf("memory is not allocated\n");

return 0;

}

for(int i=0; i<4;i++)

{

printf("enter the details of the students:%d\n",i+1);

printf("Name:\n");

scanf("%s", ptr[i].name);

printf("Mobile Number:\n");

scanf("%lld", &ptr[i].mobile);

printf("Year of birth:\n");

scanf("%d", &ptr[i].yearOfBirth);

}

printf("Student information:\n");

for (int i = 0; i < 4; i++)

{

printf("Name :%s\n",ptr[i].name);

printf("Mobile number :%lld\n",ptr[i].mobile);

printf("Year of birth :%d\n",ptr[i].yearOfBirth);

}

free(ptr);

}