**KPIT TECHNOLOGIES**

**WEEKLY REPORT**

**WEEK 2- Report (DATE: 30/5/2024)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Student name** | **Week** | **Branch** | **USN** |
| **Sunayan Dutta** | **2** | **CS (CSE)** | **1NH20CS220** |

**Yashavant Kanetkar Book**

**Question 51-100:**

**Question 51:** **Write a program to print all prime numbers from 1 to 300.**

Sol.

# include<stdio.h>

int main( )

{

int i, n = 1 ;

printf ( "\nPrime numbers between 1 and 300 are :\n1\t" ) ;

for ( n = 1 ; n <= 300 ; n++ )

{

i = 2 ;

for ( i = 2 ; i < n ; i++ )

{

if ( n % i == 0 )

break ;

}

if ( i == n )

printf ( "%d\t", n ) ;

}

return 0 ;

}

**Question 52: Write a program to add first seven terms of the following series using a for loop.**

**1/ 1!+ 2 /2! +3 /3! ……**

Sol.

# include<stdio.h>

int main( )

{

int i = 1, j ;

float fact, sum = 0.0 ;

for ( i = 1 ; i <= 7 ; i++ )

{

fact = 1.0 ;

for ( j = 1 ; j <= i ; j++ )

fact = fact \* j ;

sum = sum + i / fact ;

}

printf ( "Sum of series = %f\n", sum ) ;

return 0 ;

}

**Question 53: Write a program to generate all combinations of 1, 2 and 3 using for loop.**

Sol.

# include<stdio.h>

int main( )

{

int i = 1, j = 1, k = 1 ;

for ( i = 1 ; i <= 3 ; i++ )

{

for ( j = 1 ; j <= 3 ; j++ )

{

for ( k = 1 ; k <= 3 ; k++ )

printf ( "%d %d %d\n", i , j , k ) ;

}

}

return 0 ;

}

**Question 54: Answer the following questions:**

**(a) The break statement is used to exit from:**

**1. An if statement**

**2. A for loop**

**3. A program**

**4. The main( ) function**

**(b) A do-while loop is useful when we want that the statements within the loop must be executed:**

**1. Only once**

**2. At least once**

**3. More than once**

**4. None of the above**

**(c) In what sequence the initialization, testing and execution of body is done in a do-while loop?**

**1. Initialization, execution of body, testing**

**2. Execution of body, initialization, testing**

**3. Initialization, testing, execution of body**

**4. None of the above**

**(d) Which of the following is not an infinite loop?**

**1. int i = 1 ; while ( 1 ) { i++ ; }**

**2. for ( ; ; ) ;**

**3. int t = 0, f ; while ( t ) { f = 1 ;**

**4. int y, x = 0 ; do { y = x ;**

**(e) Which of the following statements is true for the following program?**

**# include<stdio.h>**

**int main( )**

**{**

**int x = 10, y = 100 % 90 ;**

**for ( i = 1 ; i <= 10 ; i++ ) ;**

**if ( x != y ) ;**

**printf ( "x = %d y = %d\n", x, y ) ;**

**return 0 ;**

**}**

**1. The printf( ) function is called 10 times.**

**2. The program will produce the output x = 10 y = 10.**

**3. The ; after the if ( x != y ) will not produce an error.**

**4. The program will not produce any output.**

**5. The printf( ) function is called infinite times.**

**(f) Which of the following statement is true about a for loop used in a C program?**

**1. for loop works faster than a while loop.**

**2. All things that can be done using a for loop can also be done using a while loop.**

**3. for ( ; ; ) implements an infinite loop.**

**4. for loop can be used if we want statements in a loop to get executed at least once.**

**5. for loop works faster than a do-while loop.**

Sol.

1. 2. A for loop
2. 2. At least once
3. 1. Initialization, execution of body, testing
4. 3. int t = 0, f ; while ( t ) { f = 1 ;
5. 3. The ; after the if ( x != y ) will not produce an error.
6. 2. All things that can be done using a for loop can also be done using a while loop.

**Question 55: Attempt the following questions:**

1. **Write a program to print the multiplication table of the number entered by the user. The table should get displayed in the following form: 29 \* 1 = 29 29 \* 2 = 58 …**
2. **According to a study, the approximate level of intelligence of a person can be calculated using the following formula: i = 2 + ( y + 0.5 x )**

**Write a program that will produce a table of values of I,y and x, where y varies from 1 and 6, and for each value of y , x varies from 5.5 to 12.5 in steps of 0.5.**

1. **When interest compounds q times per year at an annual rate of r% for n years, the prinicipal p compounds to an amount a as per the following formula.**

**a = p (1+r/q)nq**

**Write a program to read 10 sets of p, r, n & q and calculate the corresponding as.**

1. **The natural logarithm can be approximated by the following series.**

**(x-1)/x + ½(x-1/x)2 + ½(x-1/x)3  + ½(x-1/x)4 + ….**

**If x is input through the keyboard, write a program to calculate the sun of first seven terms of this series.**

1. **Write a program to generate all Pythagorean Triplets with side length less than or equal to 30.**
2. **Population of a town today is 100000. The population has increased steadily at the rate of 10% per year for last 10 years. Write a program to determine the population at the end of each year in the last decade.**
3. **Ramanujan number (1729) is the smallest number that can be expressed as sum of two cubes in two different ways – 1729 can be expressed as 13 + 123 . Write a program to print all such number up to a reasonable limit.**
4. **Write a program to print 24 hours of day with suitable suffixes like AM, PM, Noon and Midlight.**
5. **Write a program to produce the following output:**

**1**

**2 3**

**4 5 6**

**7 8 9 10**

Sol.

(a)

#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;

}

(b)

#include<stdio.h>

int main()

{

int y;

float x,i;

printf("yx = 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;

}

c>

#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\*b;

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

}

return 0;

}

d>

#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;

}

e>

#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 pythogorian triplets.\n");

return 0;

}

f>

#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;

}

g>

#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;

for (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;

}

h>

#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;

}

i>

#include <stdio.h>

int main()

{

int rows, i, j, number = 1;

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

scanf("%d", &rows);

for (i = 1; i <= rows; i++)

{

for (j = 1; j <= i; ++j)

{

printf("%d ", number);

++number;

}

printf("\n");

}

return 0;

}

**Question 56: Write a menu driven program which has following options:**

**1. Factorial of a number**

**2. Prime or not**

**3. Odd or even**

**4. Exit**

**Once a menu item is selected the appropriate action should be taken and once this action is finished, the menu should reappear. Unless the user selects the ‘Exit’ option the program should continue to work.**

Sol.

# include<stdio.h>

# include<stdlib.h>

int main( )

{

int choice, num, i, fact ;

while ( 1 )

{

printf ( "\n1. Factorial\n" ) ;

printf ( "2. Prime\n" ) ;

printf ( "3. Odd / Even\n" ) ;

printf ( "4. Exit\n" ) ;

printf ( "Your choice? " ) ;

scanf ( "%d", &choice ) ;

switch ( choice )

{

case 1 :

printf ( "\nEnter number: " ) ;

scanf ( "%d", &num ) ;

fact = 1 ;

for ( i = 1 ; i <= num ; i++ )

fact = fact \* i ;

printf ( "Factorial value = %d\n", fact ) ;

break ;

case 2 :

printf ( "\nEnter number: " ) ;

scanf ( "%d", &num ) ;

for ( i = 2 ; i < num ; i++ )

{

if ( num % i == 0 )

{

printf ( "Not a prime number\n" ) ;

break ;

}

}

if ( i == num )

printf ( "Prime number\n" ) ;

break ;

case 3 :

printf ( "\nEnter number: " ) ;

scanf ( "%d", &num ) ;

if ( num % 2 == 0 )

printf ( "Even number\n" ) ;

else

printf ( "Odd number\n" ) ;

break ;

case 4 :

exit ( 0 ) ;

default :

printf ( "Wrong choice!\a\n" ) ;

}

}

return 0 ;

}

**Question 57: [A] What will be the output of the following programs?**

**(a)**

**# include<stdio.h>**

**int main( )**

**{**

**char suite = 3 ;**

**switch ( suite )**

**{**

**case 1 :**

**printf ( "Diamond\n" ) ;**

**case 2 :**

**printf ( "Spade\n" ) ;**

**default :**

**printf ( "Heart\n" ) ;**

**}**

**printf ( "I thought one wears a suite\n" ) ;**

**return 0 ;**

**}**

**(b)**

**# include<stdio.h>**

**int main( )**

**{**

**int c = 3 ;**

**switch ( c )**

**{**

**case '3' :**

**printf ( "You never win the silver prize\n" ) ;**

**break ;**

**case 3 :**

**printf ( "You always lose the gold prize\n" ) ;**

**break ;**

**default :**

**printf ( "Of course provided you win a prize\n" ) ;**

**}**

**return 0 ;**

**}**

**(c)**

**# include<stdio.h>**

**int main( )**

**{**

**int i = 3 ;**

**switch ( i )**

**{**

**case 0 :**

**printf ( "Customers are dicey\n" ) ;**

**case 1 + 2 :**

**printf ( "Markets are pricey\n" ) ;**

**case 4 / 2 :**

**printf ( "Investors are moody\n" ) ;**

**}**

**return 0 ;**

**}**

**(d)**

**# include<stdio.h>**

**int main( )**

**{**

**int k ;**

**float j = 2.0 ;**

**switch ( k = j + 1 )**

**{**

**case 3 :**

**printf ( "Trapped\n" ) ;**

**break ;**

**default :**

**printf ( "Caught!\n" ) ;**

**}**

**return 0 ;**

**}**

**(e)**

**# include<stdio.h>**

**int main( )**

**{**

**int ch = 'a' + 'b' ;**

**switch ( ch )**

**{**

**case 'a' :**

**case 'b' :**

**printf ( "Look at 10 ideas, 11th will occur to you\n" ) ;**

**case 'A' :**

**printf ( "If you have a good idea, project it\n" ) ;**

**case 'b' + 'a' :**

**printf ( "Have ideas, will fly\n" ) ;**

**}**

**return 0 ;**

**}**

Sol.

(A)

Heart

I thought one wears a suite

(B)

You always lose the gold prize

(C)

Markets are pricey

Investors are moody

(D)

Trapped

(E)

Have ideas, will fly

**Question 58: [B] Point out the errors, if any, in the following programs:**

**(a)**

**# include<stdio.h>**

**int main( )**

**{**

**int suite = 1 ;**

**switch ( suite ) ;**

**{**

**case 0 ;**

**printf ( "Club\n" ) ;**

**case 1 ;**

**printf ( "Diamond\n" ) ;**

**}**

**return 0 ;**

**}**

**(b)**

**# include<stdio.h>**

**int main( )**

**{**

**int temp ;**

**scanf ( "%d", &temp );**

**switch ( temp )**

**{**

**case ( temp <= 20 ) :**

**printf ( "Ooooooohhhh! Damn cool!\n" ) ;**

**case ( temp > 20 && temp <= 30 ) :**

**printf ( "Rain rain here again!\n" ) ;**

**case ( temp > 30 && temp <= 40 ) :**

**printf ( "Wish I am on Everest\n" ) ;**

**default :**

**printf ( "Good old Nagpur weather\n" ) ;**

**}**

**return 0 ;**

**}**

**(c)**

**# include<stdio.h>**

**int main( )**

**{**

**float a = 3.5 ;**

**switch ( a )**

**{**

**case 0.5 :**

**printf ( "The art of C\n" ) ;**

**break ;**

**case 1.5 :**

**printf ( "The spirit of C\n" ) ;**

**break ;**

**case 2.5 :**

**printf ( "See through C\n" ) ;**

**break ;**

**}**

**return 0 ;**

**}**

**(d)**

**# include<stdio.h>**

**int main( )**

**{**

**int a = 3, b = 4, c ;**

**c = b – a ;**

**switch ( c )**

**{**

**case 1 || 2 :**

**printf ( "God give me a chance to change things\n" ) ;**

**break ;**

**case a || b :**

**printf ( "God give me a chance to run my show\n" ) ;**

**break ;**

**}**

**return 0 ;**

**}**

Sol.

(A)

# include<stdio.h>

int main( )

{

int suite = 1 ;

switch ( suite )

{

case 0 :

printf ( "Club\n" ) ;

break;

case 1 :

printf ( "Diamond\n" ) ;

break;

}

return 0 ;

}

(B)

# include<stdio.h>

int main( )

{

int temp ;

scanf ( "%d", &temp );

if (temp <= 20) {

printf ( "Ooooooohhhh! Damn cool!\n" );

} else if (temp > 20 && temp <= 30) {

printf ( "Rain rain here again!\n" );

} else if (temp > 30 && temp <= 40) {

printf ( "Wish I am on Everest\n" );

} else {

printf ( "Good old Nagpur weather\n" );

}

return 0 ;

}

(C)

# include<stdio.h>

int main( )

{

float a = 3.5 ;

if (a == 0.5) {

printf ( "The art of C\n" ) ;

} else if (a == 1.5) {

printf ( "The spirit of C\n" ) ;

} else if (a == 2.5) {

printf ( "See through C\n" ) ;

}

return 0 ;

}

(D)

# include<stdio.h>

int main( )

{

int a = 3, b = 4, c ;

c = b - a ;

switch ( c )

{

case 1:

case 2:

printf ( "God give me a chance to change things\n" ) ;

break ;

case 3:

case 4:

printf ( "God give me a chance to run my show\n" ) ;

break ;

}

return 0 ;

}

**Question 59 : [C] Write a program to find the grace marks for a student using switch.**

**The user should enter the class obtained by the student and the number of subjects he has failed in. Use the following logic:**

* **If the student gets first class and he fails in more than 3 subjects, he does not get any grace. Otherwise, he gets a grace of 5 marks per subject.**
* **If the student gets second class and he fails in more than 2 subjects, he does not get any grace. Otherwise, he gets a grace of 4 marks per subject.**
* **If the student gets third class and he fails in more than 1 subject, then he does not get any grace. Otherwise, he gets a grace of 5 marks.**

Sol.

#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;

}

**Question 60 : Write a function to calculate the factorial value of any integer entered through the keyboard.**

Sol.

# include<stdio.h>

int fact ( int ) ;

int main( )

{

int num, factorial ;

printf ( "\nEnter a number: " ) ;

scanf ( "%d", &num ) ;

factorial = fact ( num ) ;

printf ( "Factorial of %d = %ld\n", num, factorial ) ;

return 0 ;

}

int fact ( int num )

{

int i ;

int factorial = 1 ;

for ( i = 1 ; i <= num ; i++ )

factorial = factorial \* i ;

return ( factorial ) ;

}

**Question 61 : Write a function power ( a, b ), to calculate the value of a raised to b.**

Sol.

# include<stdio.h>

float power ( float, int ) ;

int main( )

{

float x, pow ;

int y ;

printf ( "\nEnter two numbers: " ) ;

scanf ( "%f %d", &x, &y ) ;

pow = power ( x , y ) ;

printf ( "%f to the power %d = %f\n", x, y, pow ) ;

return 0 ;

}

float power ( float x, int y )

{

int i ;

float p = 1 ;

for ( i = 1 ; i <= y ; i++ )

p = p \* x ;

return ( p ) ;

}

**Question 62 : Define a function to convert any given year into its Roman equivalent. Use these Roman equivalents for decimal numbers: 1 – I, 5 – V, 10 – X, 50 – L, 100 – C, 500 – D, 1000 – M. Example:**

**Roman equivalent of 1988 is mdcccclxxxviii.**

**Roman equivalent of 1525 is mdxxv.**

Sol.

# include<stdio.h>

int romanise ( int, int, char ) ;

int main( )

{

int yr ;

printf ( "\nEnter year: " ) ;

scanf ( "%d", &yr ) ;

yr = romanise ( yr, 1000, 'm' ) ;

yr = romanise ( yr, 500, 'd' ) ;

yr = romanise ( yr, 100, 'c' ) ;

yr = romanise ( yr, 50, 'l' ) ;

yr = romanise ( yr, 10, 'x' ) ;

yr = romanise ( yr, 5, 'v' ) ;

romanise ( yr, 1, 'i' ) ;

return 0 ;

}

int romanise ( int y, int k, char ch )

{

int i, j ; j = y / k ;

for ( i = 1 ; i <= j ; i++ )

printf ( "%c", ch ) ;

return ( y % k ) ;

}

**Question 63 : [A] Point out the errors, if any, in the following programs:**

1. **# include<stdio.h>**

**int addmult ( int, int )**

**int main( )**

**{**

**int i = 3, j = 4, k, l ;**

**k = addmult ( i, j ) ;**

**l = addmult ( i, j ) ;**

**printf ( "%d %d\n", k, l ) ;**

**return 0 ;**

**}**

**int addmult ( int x, int y )**

**{**

**int u, v ; u = x + y ;**

**v = x \* y ;**

**return ( u, v ) ;**

**}**

1. **# include<studio.h>**

**int main( )**

**{**

**int a ; a = message( ) ;**

**return 0 ;**

**}**

**void message( )**

**{**

**printf ( "Learn from him online at ykanetkar.com\n" ) ;**

**return ;**

**}**

1. **# include<stdio.h>**

**int main( )**

**{**

**float a = 15.5 ;**

**char ch = 'C' ;**

**printit ( a, ch ) ;**

**return 0 ;**

**}**

**printit ( a, ch )**

**{**

**printf ( "%f %c\n", a, ch ) ;**

**}**

1. **# include<stdio.h>**

**int main( )**

**{**

**let\_us\_c( )**

**{**

**printf ( "Learn C online…\n" ) ;**

**printf ( "At ykanetkar.com\n" ) ;**

**}**

**return 0 ;**

**}**

Sol.

(A)

# include<stdio.h>

int addmult ( int, int );

int main( )

{

int i = 3, j = 4, k, l ;

k = addmult ( i, j ) ;

l = addmult ( i, j ) ;

printf ( "%d %d\n", k, l ) ;

return 0 ;

}

int addmult ( int x, int y )

{

int u, v ;

u = x + y ;

v = x \* y ;

return u + v ;

}

(B)

# include<stdio.h>

void message( );

int main( )

{

int a ;

message( );

return 0 ;

}

void message( )

{

printf ( "Learn from him online at ykanetkar.com\n" ) ;

return ;

}

(C)

# include<stdio.h>

void printit ( float a, char ch );

int main( )

{

float a = 15.5 ;

char ch = 'C' ;

printit ( a, ch ) ;

return 0 ;

}

void printit ( float a, char ch )

{

printf ( "%f %c\n", a, ch ) ;

}

(D)

# include<stdio.h>

void let\_us\_c( );

int main( )

{

let\_us\_c( );

return 0 ;

}

void let\_us\_c( )

{

printf ( "Learn C online…\n" ) ;

printf ( "At ykanetkar.com\n" ) ;

}

**Question 64 : [B] State whether the following statements are True or False:**

**(a) Commonly used variables are available to all the functions in a program.**

**(b) To return the control back to the calling function we must use the keyword return.**

**(c) The same variable names can be used in different functions without any conflict.**

**(d) Every called function must contain a return statement.**

**(e) A function may contain more than one return statement.**

**(f) Each return statement in a function may return a different value.**

**(g) A function can still be useful even if you don’t pass any arguments to it and the function doesn’t return any value.**

**(h) Same names can be used for different functions without any conflict.**

**(i) A function may be called more than once from any other function.**

Sol.

(a) False

(b) True

(c) True

(d) False

(e) True

(f) True

(g) True

(h) False

(i) True

**Question 64 : [C] Answer the following questions:**

**(a) Any year is entered through the keyboard. Write a function to determine whether the year is a leap year or not.**

**(b) A positive integer is entered through the keyboard. Write a function to obtain the prime factors of this number.**

**For example, prime factors of 24 are 2, 2, 2 and 3, whereas prime factors of 35 are 5 and 7.**

Sol.

(a)

#include<stdio.h>

int calc(int year)

{

if(yr % 4 == 0 && (yr % 100 != 0 || yr % 400 == 0))

return 1;

else

return 0;

}

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;

}

(b)

#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;

}

**Question 65 : Write a function that receives 5 integers and returns the sum, average and standard deviation of these numbers. Call this function from main( ) and print the results in main( ).**

Sol.

# include<stdio.h>

# include<math.h>

void stats ( int \*, int \*, double \* ) ;

int main( )

{

int sum, avg ;

double stdev ;

stats ( &sum, &avg, &stdev ) ;

printf ( "Sum = %d \nAverage = %d \nStandard deviation = %lf\n", sum, avg, stdev ) ;

return 0 ;

}

void stats ( int \*sum, int \*avg, double \*stdev )

{

int n1, n2, n3, n4, n5;

printf ( "\nEnter 5 numbers: " ) ;

scanf ( "%d%d%d%d%d", &n1, &n2, &n3, &n4, &n5 ) ;

\*sum = n1 + n2 + n3 + n4 + n5 ;

\*avg = \*sum / 5 ;

\*stdev = sqrt ( ( pow ( ( n1 - \*avg ), 2.0 ) + pow ( ( n2 - \*avg ), 2.0 ) + \ pow ( ( n3 - \*avg ), 2.0 ) + pow ( ( n4 - \*avg ), 2.0 ) + \ pow ( ( n5 - \*avg ), 2.0 ) ) / 4 ) ;

}

**Question 66 : Write a program that defines a function that calculates power of one number raised to another and factorial value of a number in one call.**

Sol.

# include<stdio.h>

void power\_fact ( float, int, int, float \*, int \* ) ;

int main( )

{

float a ;

int b, number, factorial ;

float pow ;

printf ( "Enter a and b for calculating a raised to b: " ) ;

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

printf ( "Enter number whose factorial is to be calculated: " ) ;

scanf ( "%d", &number ) ;

power\_fact ( a, b, number, &pow, &factorial ) ;

printf ( "Power = %f Factorial = %d", pow, factorial ) ;

return 0 ;

}

void power\_fact ( float x, int y, int num, float \*power, int \*fact )

{

float res = 1 ;

int i ;

for ( i = 1 ; i <= y ; i++ )

res = res \* x ;

\*power = res ;

res = 1 ;

for ( i = 1 ; i <= num ; i++ )

res = res \* i ;

\*fact = res ;

}

**Question 67 : Figure 9.4 shows three memory locations and values stored in them. Write a program to declare variables that implement the relationship shown. How will you print the values and addresses shown in the figure? On which machine the program should be executed to get such addresses?**

Sol.

#include<stdio.h>

int main( )

{

float x = 3.14 ;

float \*y ;

float \*\*z ;

= &x ;

z = &y ;

printf ( "%p %p %p\n", &x, &y, &z ) ;

printf ( "%p %p\n", y, z ) ;

printf ( "%f %f %f", x, \*y, \*\*z ) ;

return 0 ;

}

**Question 68 : [A] What will be the output of the following programs?**

**(a)**

**# include<stdio.h>**

**void fun ( int, int ) ;**

**int main( )**

**{**

**int i = 5, j = 2 ;**

**fun ( i, j ) ;**

**printf ( "%d %d\n", i, j ) ;**

**return 0 ;**

**}**

**void fun ( int i, int j )**

**{**

**i = i \* i ;**

**j = j \* j ;**

**}**

**(b)**

**# include<stdio.h>**

**void fun ( int \*, int \* ) ;**

**int main( )**

**{**

**int i = 5, j = 2 ;**

**fun ( &i, &j ) ;**

**printf ( "%d %d\n", i, j ) ;**

**return 0 ;**

**}**

**void fun ( int \*i, int \*j )**

**{**

**\*i = \*i \* \*i ;**

**\*j = \*j \* \*j ;**

**}**

**(c)**

**# include<stdio.h>**

**int main( )**

**{**

**float a = 13.5 ;**

**float \*b, \*c ; b = &a ;**

**c = b ;**

**printf ( "%u %u %u\n", &a, b, c ) ;**

**printf ( "%f %f %f %f %f\n", a, \*(&a), \*&a, \*b, \*c ) ;**

**return 0 ;**

**}**

Sol.

(A)

5 2

(B)

25 4

(C)

4160208 4160208 4160208 // (Note: Actual addresses will vary)

13.500000 13.500000 13.500000 13.500000 13.500000

**Question 69 : [B] Point out the errors, if any, in the following programs:**

**(a)**

**# include<stdio.h>**

**int main( )**

**{**

**float p = 23.24 ;**

**int \*q, \*\*r ;**

**q = &p ;**

**r = &q ;**

**printf ( "%f %f\n", \*q, \*\*r ) ;**

**return 0 ;**

**}**

**(b)**

**# include<stdio.h>**

**int main( )**

**{**

**char ch = 'A' ;**

**int k = 35 ;**

**float a = 3.14 ;**

**printf ( "%p %p %p\n", &ch, &k, &a ) ;**

**return 0 ;**

**}**

**(c)**

**# include<stdio.h>**

**void function ( int \* ) ;**

**int main( )**

**{**

**int i = 35, \*z ;**

**z = function ( &i ) ;**

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

**return 0 ;**

**}**

**void function ( int \*m )**

**{**

**return ( \*m + 2 ) ;**

**}**

(A)

# include<stdio.h>

int main( )

{

float p = 23.24 ;

float \*q;

float \*\*r;

q = &p ;

r = &q ;

printf ( "%f %f\n", \*q, \*\*r ) ;

return 0 ;

}

(B)

# include<stdio.h>

int main( )

{

char ch = 'A' ;

int k = 35 ;

float a = 3.14 ;

printf ( "%p %p %p\n", (void\*)&ch, (void\*)&k, (void\*)&a ) ;

return 0 ;

}

(C)

# include<stdio.h>

void function ( int \* ) ;

int main( )

{

int i = 35;

function ( &i ) ;

printf ( "%d\n", i ) ;

return 0 ;

}

void function ( int \*m )

{

\*m = \*m + 2 ;

}

**Question 70 : [C] Attempt the following questions:**

**(a) Given three variables x, y, z, write a function to circularly shift their values to right. In other words, if x = 5, y = 8, z = 10, after circular shift y = 5, z = 8, x =10. Call the function with variables a, b, c to circularly shift values.**

**(b) Define a function that receives weight of a commodity in Kilograms and returns the equivalent weight in Grams, Tons and Pounds. Call this function from main( ) and print the results in main( ).**

**(c) Define a function to compute the distance between two points and use it to develop another function that will compute the area of the triangle whose vertices are A(x1, y1), B(x2, y2), and C(x3, y3). Use these functions to develop a function which returns a value 1 if the point (x, y) lines inside the triangle ABC, otherwise returns a value 0. Would you get any advantage if you develop these functions to work on call be reference principle?**

Sol.

a>

#include<stdio.h>

#include<conio.h>

void chng(int \*a, int \*b, int \*c)

{

int x;

x = \*c;

\*c = \*b;

\*b = \*a;

\*a = x;

}

int main()

{

int a, b, c;

printf("Enter three numbers : ");

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

printf("\n\nYou've Entered\n\na : %d b : %d c : %d\n", a, b, c);

chng(&a, &b, &c);

printf("\nAfter Shifting\n\na : %d b : %d c : %d\n", a, b, c);

getch();

return 0;

}

b>

#include<stdio.h>

int weight(float \*g,float \*t,float \*l)

{

float kg;

printf("Enter the weight of a commodity in kgs:");

scanf("%f",&kg);

g=1000\*kg;

t=0.001\*kg;

l=2.25\*kg;

}

int main()

{

float a,b,c;

weight(&a,&b,&c);

printf("Gram:%f",a);

printf("Tons:%f",b);

printf("Pounds:%f",c);

return 0;

}

c>

#include<stdio.h>

#include<math.h>

void res(float x1, float y1, float x2, float y2, float x3, float y3,

float x, float y, float \*area, int \*flag);

float distance(float x1, float y1, float x2, float y2);

float cal\_area(float a, float b, float c);

int position(float area, float A, float B, float C);

void main()

{

float x1, y1, x2, y2, x3, y3, x, y;

float area=0;

int flag=0;

printf("Enter the values of (x1,y1): ");

scanf("%f%f", &x1, &y1);

printf("Enter the values of (x2,y2): ");

scanf("%f%f", &x2, &y2);

printf("Enter the values of (x3,y3): ");

scanf("%f%f", &x3, &y3);

printf("Enter point values (x,y): ");

scanf("%f%f", &x, &y);

res(x1, y1, x2, y2, x3, y3, x, y, &area, &flag);

printf("\n Area of triangle: %.2f\n", area);

if (flag==1)

printf("\nPoint(%.2f, %.2f) lies inside the triangle.",x,y);

else

printf("\nPoint(%.2f, %.2f) lies outside the triangle.",x,y);

}

void res(float x1, float y1, float x2, float y2, float x3, float y3,

float x, float y, float \*area, int \*flag)

{

float a, b, c, d, e, f, A, B, C ;

a = distance(x1,y1,x2,y2);

b = distance(x2,y2,x3,y3);

c = distance(x3,y3,x1,y1);

\*area = cal\_area(a,b,c);

d = distance(x1,y1,x,y);

e = distance(x2,y2,x,y);

f = distance(x3,y3,x,y);

A = cal\_area(d,e,a);

B = cal\_area(e,b,f);

C = cal\_area(f,c,d);

\*flag = position(\*area, A, B, C);

}

float distance(float x1, float y1, float x2, float y2)

{

return (sqrt(pow((x2-x1),2)+pow((y2-y1),2)));

}

float cal\_area(float a, float b, float c)

{

float S,ar;

S = (a+b+c)/2.0;

ar = (sqrt(S\*(S-a)\*(S-b)\*(S-c)));

return (ar);

}

int position(float area, float A, float B, float C)

{

float res;

res = area-(A+B+C);

if(res==0 || res<0.0001)

{

return(1);

}

else

{

return(0);

}

}

**Question 71 : A 5-digit positive integer is entered through the keyboard, write a recursive function to calculate sum of digits of the 5-digit number.**

Sol.

# include<stdio.h>

int rsum ( int ) ;

int main( )

{

int num, sum ;

int n ;

printf ( "Enter number: " ) ;

scanf ( "%d", &num ) ;

sum = rsum ( num ) ;

printf ( "Sum of digits is %d\n", sum ) ;

return 0 ;

}

int rsum ( int n )

{

int s, remainder ;

if ( n != 0 )

{

remainder = n % 10 ;

s = remainder + rsum ( n / 10 ) ;

}

else

return 0 ;

return s ;

}

**Question 72 : A positive integer is entered through the keyboard, write a program to obtain the prime factors of the number. Modify the function suitably to obtain the prime factors recursively.**

Sol.

# include<stdio.h>

void factorize ( int, int ) ;

int main( )

{

int num ;

printf ( "Enter a number: " ) ;

scanf ( "%d", &num ) ;

printf ( "Prime factors are: " ) ;

factorize ( num, 2 ) ;

return 0 ;

}

void factorize ( int n, int i )

{

if ( i <= n )

{

if ( n % i == 0 )

{

printf ( "%d ", i ) ;

n = n / i ;

}

else

i++ ;

factorize ( n, i ) ;

}

}

**Question 73 : Write a recursive function to obtain the first 25 numbers of a Fibonacci sequence. In a Fibonacci sequence the sum of two successive terms gives the third term. Following are the first few terms of the Fibonacci sequence :**

**0 1 1 2 3 5 8 13 21 34 55 89….**

Sol.

#include<stdio.h>

int fibo ( int ) ;

int main( )

{

int terms = 25, i, n = 0 ;

for ( i = 1 ; i <= terms ; i++ )

{

printf ( "%d\t", fibo ( n ) ) ;

n++ ;

}

return 0 ;

}

int fibo ( int n )

{

if ( n == 0 || n == 1 )

return n ;

else

return ( fibo ( n - 1 ) + fibo ( n - 2 ) ) ;

}

**Question 74 : [A] What will be the output of the following programs?**

**(a)**

**# include<stdio.h>**

**int main( )**

**{**

**printf ( "I C, you C, we all C\n" ) ;**

**main( ) ;**

**return 0 ;**

**}**

**(b)**

**# include<stdio.h>**

**# include<stdlib.h>**

**int main( )**

**{**

**int i = 0 ;**

**i++ ;**

**if ( i <= 5 )**

**{**

**printf ( "C adds wings to your thoughts\n" ) ;**

**exit ( 0 ) ;**

**main( ) ;**

**}**

**return 0 ;**

**}**

Sol.

a)

I C, you C, we all C

I C, you C, we all C

I C, you C, we all C

...

b)

C adds wings to your thoughts

**Question 75 : [B] Attempt the following questions:**

**(a) A positive integer is entered through the keyboard, write a function to find the binary equivalent of this number:**

**(1) Without using recursion**

**(2) Using recursion**

**(b) Write a recursive function to obtain the sum of first 25 natural numbers.**

**(c) There are three pegs labeled A, B and C. Four disks are placed on peg A. The bottom-most disk is largest, and disks go on decreasing in size with the topmost disk being smallest. The objective of the game is to move the disks from peg A to peg C, using peg B as an auxiliary peg. The rules of the game are as follows:**

**(1) Only one disk may be moved at a time, and it must be the top disk on one of the pegs.**

**(2) A larger disk should never be placed on the top of a smaller disk.**

**Write a program to print out the sequence in which the disks should be moved such that all disks on peg A are finally transferred to peg C.**

Sol.

a>

#include<stdio.h>

void nonrec\_bin(int num)

{

int sum = 0;

for (; num; num /= 2)

{

sum += num % 2;

sum \*= 10;

}

printf("%d", sum);

}

int bin(int number)

{

if(number>1)

{

bin(number/2);

}

printf("%d",number%2);

return 0;

}

int main()

{

int number;

printf("Enter number\n");

scanf("%d",&number);

nonrec\_bin(number);

bin(number);

return 0;

}

b>

#include<stdio.h>

int sum(int x)

{

if (x == 0)

return x;

x = x + sum(x - 1);

return x;

}

int main()

{

int number,ans;

printf("Enter number\n");

scanf("%d",&number);

ans=sum(number);

printf("%d",ans);

return 0;

}

c>

#include <stdio.h>

void towerOfHanoi(int n, char from\_rod, char to\_rod, char aux\_rod)

{

if (n == 1)

{

printf("\n Move disk 1 from rod %c to rod %c", from\_rod, to\_rod);

return;

}

towerOfHanoi(n-1, from\_rod, aux\_rod, to\_rod);

printf("\n Move disk %d from rod %c to rod %c", n, from\_rod, to\_rod);

towerOfHanoi(n-1, aux\_rod, to\_rod, from\_rod);

}

int main()

{

int n = 4;

towerOfHanoi(n, 'A', 'C', 'B');

return 0;

}

**Question 76 : Write macro definitions for the following:**

**1. To test whether a character is a lowercase letter or not.**

**2. To test whether a character is an uppercase letter or not.**

**3. To test whether a character is an alphabet or not. Make use of the macros you defined in 1 and 2 above.**

**4. To obtain the bigger of two numbers.**

Sol.

# include<stdio.h>

#define ISUPPER(x) ( x >= 65 && x <= 90 ? 1 : 0 )

#define ISLOWER(x) ( x >= 97 && x <= 122 ? 1 : 0 )

#define ISALPHA(x) ( ISUPPER(x) || ISLOWER(x) )

#define BIG(x,y) ( x > y ? x : y )

int main( )

{

char ch ;

int d, a, b ;

printf ( "\nEnter any alphabet/character: " ) ;

scanf ( "%c", &ch ) ;

if ( ISUPPER ( ch ) == 1 )

printf ( "You entered a capital letter\n" ) ;

if ( ISLOWER ( ch ) == 1 )

printf ( "You entered a small case letter\n" ) ;

if ( ISALPHA ( ch ) != 1 )

printf ( "You entered character other than an alphabet\n" ) ;

printf ( "Enter any two numbers: " ) ;

scanf ( "%d%d", &a, &b ) ; d = BIG ( a, b ) ;

printf ( "Bigger number is %d\n", d ) ;

return 0 ;

}

**Question 77 : Write macro definitions with arguments for calculation of area and perimeter of a triangle, a square and a circle. Store these macro definitions in a file “areaperi.h”. Include this file in your program, and use the macro definitions for calculating area and perimeter for different squares, triangles and circles.**

Sol.

/\* areaperi.h \*/

#define PI 3.1415

#define PERIC( r ) ( 2 \* PI \* r )

#define AREAC( r ) ( PI \* r \* r )

#define PERIS( x ) ( 4 \* x )

#define AREAS( x ) ( x \* x )

#define PERIT( x, y, z ) ( x + y + z )

#define AREAT( b, h ) ( 0.5 \* b \* h )

# include<stdio.h>

# include "areaperi.h"

int main( )

{

int d, a, b ;

float sid1, sid2, sid3, sid, p\_tri, p\_cir, p\_sqr, a\_tri, a\_cir,a\_sqr ;

float r, base, height ;

printf ( "\nEnter radius of circle: " ) ;

scanf ( "%f", &r ) ; p\_cir = PERIC ( r ) ;

printf ( "Circumference of circle = %f\n", p\_cir ) ;

a\_cir = AREAC ( r ) ;

printf ( "Area of circle = %f\n", a\_cir ) ;

printf ( "Enter side of a square: " ) ;

scanf ( "%f", &sid ) ; p\_sqr = PERIS ( sid ) ;

printf ( "Perimeter of square = %f\n", p\_sqr ) ;

a\_sqr = AREAS ( sid ) ;

printf ( "Area of square = %f\n", a\_sqr ) ;

printf ( "Enter length of 3 sides of triangle: " ) ;

scanf ( "%f %f %f", &sid1, &sid2, &sid3 ) ;

p\_tri = PERIT ( sid1, sid2, sid3 ) ;

printf ( "Perimeter of triangle = %f\n", p\_tri ) ;

printf ( "Enter base and height of triangle: " ) ;

scanf ( "%f %f", &base, &height ) ;

a\_tri = AREAT ( base, height ) ;

printf ( "Area of triangle = %f\n", a\_tri ) ;

return 0 ;

}

**Question 78 : [A] Answer the following questions:**

**(a) A preprocessor directive is:**

**1. A message from compiler to the programmer**

**2. A message from compiler to the linker**

**3. A message from programmer to the preprocessor**

**4. A message from programmer to the microprocessor**

**(b) Which of the following are correctly formed #define statements?**

**#define INCH PER FEET 12**

**#define SQR (X) ( X \* X )**

**#define SQR(X) X \* X**

**#define SQR(X) ( X \* X )**

**(c) State True or False:**

**1. A macro must always be written in capital letters.**

**2. A macro must always be accommodated in a single line.**

**3. After preprocessing when the program is sent for compilation the macros are removed from the expanded source code.**

**4. Macros with arguments are not allowed.**

**5. In a macro call the control is passed to the macro.**

**(d) A header file is:**

**1. A file that contains standard library functions**

**2. A file that contains function declarations and macros**

**3. A file that contains user-defined functions**

**4. A file that is present in current working directory**

**(e) All macro substitutions in a program are done:**

**1. Before compilation of the program**

**2. After compilation of the program**

**3. During execution of the program**

**4. During linking of the program**

Sol.

1. 3. A message from programmer to the preprocessor
2. #define SQR(X) ( X \* X )
3. 1. **False**

**2.** **False**

**3. True**

**4. False**

**5. False**

1. 2. A file that contains function declarations and macros
2. 1. Before compilation of the program

**Question 79 : [B] What will be the output of the following programs?**

1. **# include<stdio.h>**

**int main( )**

**{**

**int i = 2 ;**

**# ifdef DEF**

**i \*= i ;**

**# else**

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

**# endif**

**return 0 ;**

**}**

1. **# include<stdio.h>**

**# define PRODUCT(x) ( x \* x )**

**int main( )**

**{**

**int i = 3, j, k, l ;**

**j = PRODUCT( i + 1 ) ;**

**k = PRODUCT( i++ ) ;**

**l = PRODUCT ( ++i ) ;**

**printf ( "%d %d %d %d\n", i, j, k, l ) ;**

**return 0 ;**

**}**

1. **# include<stdio.h>**

**# define PI 3.14**

**# define AREA( x, y, z ) ( PI \* x \* x + y \* z ) ;**

**int main( )**

**{**

**float a = AREA ( 1, 5, 8 ) ;**

**float b = AREA ( AREA ( 1, 5, 8 ), 4, 5 ) ;**

**printf ( " a = %f\n", a ) ;**

**printf ( " b = %f\n", b ) ;**

**return 0 ;**

**}**

Sol.

1. 2
2. 5 7 12 25
3. a = 43.140000

b = 5858.221680

**Question 80 : [C] Attempt the following questions:**

**(a) If a macro is not getting expanded as per your expectation, how will you find out how is it being expanded by the preprocessor?**

**(b) Write macro definitions for the following:**

**1. To find arithmetic mean of two numbers.**

**2. To find absolute value of a number.**

**3. To convert an uppercase alphabet to lowercase.**

**4. To obtain the biggest of three numbers.**

**(c) Write macro definitions with arguments for calculation of Simple Interest and Amount. Store these macro definitions in a file “interest.h”. Include this file in your program, and use the macro definitions for calculating Simple Interest and Amount.**

Sol.

gcc -E your\_program.c -o preprocessed\_output.c

(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)

#define SI(a,b,c) ((a\*b\*c)/100.0)

#define AMOUNT(si,p) (si+p)

#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;

}

**Question 81 : Write a program that interchanges elements at odd position with elements at even position in an array of 10 elements.**

Sol.

# include<stdio.h>

int main( )

{

int num[ ] = { 12, 4, 5, 1, 9, 13, 11, 19, 54, 34 } ;

int i, t ;

for ( i = 0 ; i <= 9 ; i = i + 2 )

{

t = num[ i ] ;

num [ i ] = num [ i + 1 ] ;

num [ i + 1 ] = t ;

}

for ( i = 0 ; i <= 9 ; i++ )

printf ( "%d\t", num[ i ] ) ;

return 0 ;

}

**Question 82 : Write a program to copy the contents of a 5-element integer array into another array in reverse order.**

Sol.

# include<stdio.h>

int main( )

{

int arr1[ 5 ], arr2[ 5 ], i, j ;

printf ( "\nEnter 5 elements of array:\n" ) ;

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

scanf ( "%d", &arr1[ i ] ) ;

for ( i = 0, j = 4 ; i <= 4 ; i++, j-- )

arr2[ j ] = arr1[ i ] ;

printf ( "Elements in reverse order:\n" ) ;

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

printf ( "%d\t", arr2[ i ] ) ;

return 0 ;

}

**Question 83 : An array contains 10 integers. Receive the number to be searched in the array as input. Write a program to search this number in the array and display the number of times it occurs in the array.**

Sol.

# include<stdio.h>

int main( )

{

int num[ ] = { 7, 3, 5, 4, 6, 7, 2, 4, 6, 7 } ;

int n, i, count ;

printf ( "\nEnter an element to search: " ) ;

scanf ( "%d", &n ) ;

count = 0 ;

for ( i = 0 ; i <= 9 ; i++ )

{

if ( num[ i ] == n )

count++ ;

}

printf ( "Number %d is found %d time(s) in the array\n", n, count ) ;

return 0 ;

}

**Question 84 : [A] Answer the following questions:**

1. **Are the following array declarations correct?**

**int a (25) ;**

**int size = 10, b[ size ] ;**

1. **Which element of the array does this expression reference?**

**num[ 4 ]**

1. **What is the difference between the 5’s in these two expressions?**

**int num[ 5 ] ;**

**num[ 5 ] = 11 ;**

1. **What will happen if you try to put so many values into an array when you initialize it that the size of the array is exceeded?**
2. **What will happen if you put too few elements in an array when you initialize it?**
3. **What will happen if you assign a value to an element of an array whose subscript exceeds the size of the array?**
4. **When you pass an array as an argument to a function, what actually gets passed?**
5. **If you don’t initialize a static array, what will its elements be set to?**
6. **if int s[ 5 ] is a one-dimensional array of integers, how will you refer to the third element in the array using pointer notation?**

Sol.

1. No, this declaration is incorrect for an array. In C or C++, int a(25); is actually a declaration of an integer variable a initialized with the value 25. The correct way to declare an array with 25 elements is int a[25];.

No, this declaration is incorrect in C and C++ (before C++11). The size of the array must be a constant expression. int size = 10; declares size as a variable, not a constant. The correct way to declare an array with a size determined at runtime is using dynamic memory allocation, e.g., int\* b = new int[size]; in C++ or int\* b = (int\*)malloc(size \* sizeof(int)); in C. In modern C++ (C++11 and later), you can use constexpr to ensure size is a constant expression.

1. This expression references the fifth element of the array num.
2. In int num[5];, the 5 specifies the size of the array num, meaning it can hold 5 elements (indexed from 0 to 4).

In num[5] = 11;, the 5 is used as an index to access an element of the array. However, this is out of bounds since valid indices for num are 0 through 4. Accessing num[5] results in undefined behavior.

1. If you initialize an array with more values than its declared size, it results in a compilation error. For example, int arr[3] = {1, 2, 3, 4}; will not compile because there are four values but only three slots.
2. If you initialize an array with fewer values than its declared size, the remaining elements are initialized to zero. For example, int arr[5] = {1, 2}; results in arr being {1, 2, 0, 0, 0}.
3. Assigning a value to an element of an array with a subscript that exceeds the size of the array results in undefined behavior. This can lead to program crashes, memory corruption, or other unexpected behavior since it accesses memory outside the bounds of the array.
4. When you pass an array to a function, a pointer to the first element of the array is passed. The function does not receive the entire array, but rather the address of the first element. For example, if you pass int arr[10] to a function, the function parameter might be declared as int\* arr.
5. If a static array is not explicitly initialized, all its elements are automatically set to zero. This applies to both global and static local arrays.
6. The third element of the array s can be referred to using pointer notation as \*(s + 2). This is because array indexing s[i] is equivalent to \*(s + i).

**Question 85 : (a) Twenty-five numbers are entered from the keyboard into an array. Write a program to find out how many of them are positive, how many are negative, how many are even and how many odd.**

**(b) If an array arr contains n elements, then write a program to check if arr[ 0 ] = arr[ n - 1 ], arr[ 1 ] = arr[ n - 2 ] and so on.**

**(c) Write a program using pointers to find the smallest number in an array of 25 integers.**

Sol.

(a)

#include <stdio.h>

int main()

{

int numbers[25];

int positiveCount = 0, negativeCount = 0, evenCount = 0, oddCount = 0;

printf("Enter 25 numbers:\n");

for (int i = 0; i < 25; i++) {

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

if (numbers[i] > 0) {

positiveCount++;

} else if (numbers[i] < 0) {

negativeCount++;

}

if (numbers[i] % 2 == 0) {

evenCount++;

} else {

oddCount++;

}

}

printf("Positive numbers: %d\n", positiveCount);

printf("Negative numbers: %d\n", negativeCount);

printf("Even numbers: %d\n", evenCount);

printf("Odd numbers: %d\n", oddCount);

return 0;

}

(b)

#include <stdio.h>

int main() {

int n;

printf("Enter the number of elements in the array: ");

scanf("%d", &n);

int arr[n];

printf("Enter the elements of the array:\n");

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

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

}

int isSymmetric = 1; // Assume the array is symmetric

for (int i = 0; i < n / 2; i++) {

if (arr[i] != arr[n - 1 - i]) {

isSymmetric = 0; // Found a mismatch

break;

}

}

if (isSymmetric) {

printf("The array is symmetric.\n");

} else {

printf("The array is not symmetric.\n");

}

return 0;

}

(c)

#include <stdio.h>

int main() {

int numbers[25];

int \*ptr = numbers;

printf("Enter 25 numbers:\n");

for (int i = 0; i < 25; i++) {

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

}

int smallest = \*ptr; // Assume first number is the smallest

for (int i = 1; i < 25; i++) {

if (\*(ptr + i) < smallest) {

smallest = \*(ptr + i);

}

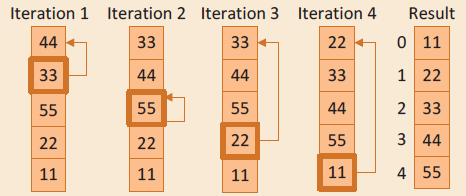
}

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

return 0;

}

**Question 86 : (d) Implement the Insertion Sort algorithm shown in Figure 13.3 on a set of 25 numbers.**

****

**(e) Write a program which performs the following tasks:**

**\*Initialize an integer array of 10 elements in main( )**

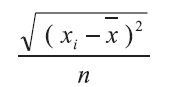
**\*Pass the entire array to a function modify( )**

**In modify() multiply each element of array by 3 Return the control to main() and print the new array elements in main()**

**(f) For the following set of sample data, compute the standard deviation and the mean.**

**-6, -12, 8, 13, 11, 6, 7, 2, 6, 9, 10, 11, 10, 9, 2**

**The formula for standard deviation is**

****

**where x, is the data item and x is the mean..**

Sol.

(d)

Iteration 1: The sorted sublist contains only the first element (22). The number 11 is removed from the unsorted sublist and inserted before 22 since 11 is smaller.

Iteration 2: The sorted sublist now contains the first two elements (11 and 22). The number 33 is removed from the unsorted sublist and inserted after 22 since 22 is smaller than 33 but larger than 11.

Iteration 3: The sorted sublist now contains the first three elements (11, 22, and 33). The number 44 is removed from the unsorted sublist and inserted after 33 since 33 is smaller than 44.

Iteration 4: The sorted sublist now contains all four elements (11, 22, 33, and 44). The number 55 is removed from the unsorted sublist and inserted at the end since it's the largest element so far.

(e)

#include <stdio.h>

void modify(int arr[], int size) {

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

arr[i] \*= 3;

}

}

int main() {

int arr[10];

printf("Enter 10 integers:\n");

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

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

}

modify(arr, 10);

printf("Modified array elements:\n");

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

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

}

printf("\n");

return 0;

}

(f)

#include <stdio.h>

#include <math.h>

int main() {

int data[] = {-6, -12, 8, 13, 11, 6, 7, 2, 6, 9, 10, 11, 10, 9, 2};

int n = sizeof(data) / sizeof(data[0]);

double sum = 0;

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

sum += data[i];

}

double mean = sum / n;

double variance = 0;

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

variance += pow(data[i] - mean, 2);

}

double std\_dev = sqrt(variance / n);

printf("Mean: %.2f\n", mean);

printf("Standard Deviation: %.2f\n", std\_dev);

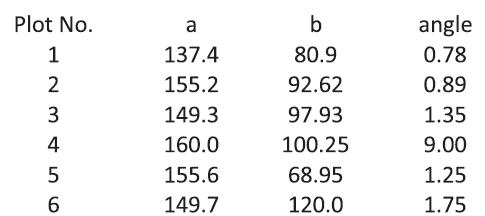
return 0;

}

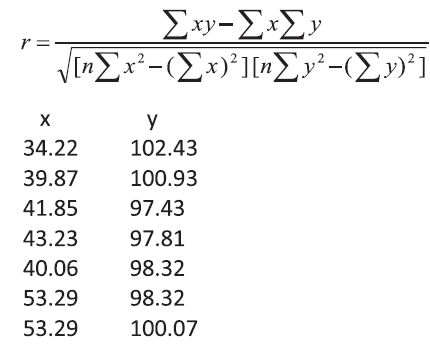
**Question 87 : (g) The area of a triangle can be computed by the sine law when 2 sides of the triangle and the angle between them are known.**

**Area (1/2) ab sin (angle)**

**Given the following 6 triangular pieces of land, write a program to find their area and determine which is largest.**

****

**(h) For the following set of n data points (x, y), write a program to compute the correlation coefficient r, given by**

****

**54.14 97.08  
49.12 91.59  
40.71 94.85  
55.15 94.65**

**(i) The X and Y coordinates of 10 different points are entered through the keyboard. Write a program to find the distance of last point from the first point (sum of distances between consecutive points).**

**(j) A dequeue is an ordered set of elements in which elements may be inserted or retrieved from either end. Using an array simulate a dequeue of characters and the operations retrieve left, retrieve right, insert left, insert right. Exceptional conditions such as dequeue full or empty should be reported. Use two pointers left and right for this simulation.**

Sol.

(g)

#include <stdio.h>

#include <math.h>

int main() {

double sides[][3] = {

{10, 12, 30}, // Triangle 1

{8, 15, 40}, // Triangle 2

{12, 12, 45}, // Triangle 3

{9, 10, 60}, // Triangle 4

{6, 10, 75}, // Triangle 5

{15, 18, 90} // Triangle 6

};

double max\_area = -1;

int max\_index = -1;

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

double a = sides[i][0];

double b = sides[i][1];

double angle = sides[i][2];

double area = 0.5 \* a \* b \* sin(angle \* M\_PI / 180); // Convert angle to radians

if (area > max\_area) {

max\_area = area;

max\_index = i + 1;

}

printf("Area of Triangle %d: %.2f\n", i + 1, area);

}

printf("Triangle %d has the largest area: %.2f\n", max\_index, max\_area);

return 0;

}

(h)

#include <stdio.h>

#include <math.h>

int main() {

double data[][2] = {

{54.14, 97.08},

{49.12, 91.59},

{40.71, 94.85},

{55.15, 94.65}

};

int n = sizeof(data) / sizeof(data[0]);

double sum\_x = 0, sum\_y = 0;

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

sum\_x += data[i][0];

sum\_y += data[i][1];

}

double mean\_x = sum\_x / n;

double mean\_y = sum\_y / n;

double numerator = 0, denominator\_x = 0, denominator\_y = 0;

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

numerator += (data[i][0] - mean\_x) \* (data[i][1] - mean\_y);

denominator\_x += pow(data[i][0] - mean\_x, 2);

denominator\_y += pow(data[i][1] - mean\_y, 2);

}

double r = numerator / sqrt(denominator\_x \* denominator\_y);

printf("Correlation Coefficient (r): %.2f\n", r);

return 0;

}

(i)

#include <stdio.h>

#include <math.h>

float distance(int x1, int y1, int x2, int y2) {

return sqrt(pow((x2 - x1), 2) + pow((y2 - y1), 2));

}

int main() {

int x[10], y[10];

printf("Enter the coordinates of 10 points:\n");

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

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

}

float total\_distance = 0;

for (int i = 1; i < 10; i++) {

total\_distance += distance(x[i - 1], y[i - 1], x[i], y[i]);

}

printf("Total distance from the first point to the last point: %.2f\n", total\_distance);

return 0;

}

(j)

#include <stdio.h>

#include <stdbool.h>

#define MAX\_SIZE 10

char dequeue[MAX\_SIZE];

int left = -1, right = -1;

bool isEmpty() {

return left == -1 && right == -1;

}

bool isFull() {

return (right + 1) % MAX\_SIZE == left;

}

void insertLeft(char value) {

if (isFull()) {

printf("Dequeue is full. Cannot insert.\n");

return;

}

if (isEmpty()) {

left = right = 0;

} else {

left = (left - 1 + MAX\_SIZE) % MAX\_SIZE;

}

dequeue[left] = value;

}

void insertRight(char value) {

if (isFull()) {

printf("Dequeue is full. Cannot insert.\n");

return;

}

if (isEmpty()) {

left = right = 0;

} else {

right = (right + 1) % MAX\_SIZE;

}

dequeue[right] = value;

}

char retrieveLeft() {

if (isEmpty()) {

printf("Dequeue is empty. Cannot retrieve.\n");

return '\0';

}

char value = dequeue[left];

if (left == right) {

left = right = -1;

} else {

left = (left + 1) % MAX\_SIZE;

}

return value;

}

char retrieveRight() {

if (isEmpty()) {

printf("Dequeue is empty. Cannot retrieve.\n");

return '\0';

}

char value = dequeue[right];

if (left == right) {

left = right = -1;

} else {

right = (right - 1 + MAX\_SIZE) % MAX\_SIZE;

}

return value;

}

int main() {

insertRight('A');

insertRight('B');

insertLeft('C');

printf("Retrieved from left: %c\n", retrieveLeft());

printf("Retrieved from right: %c\n", retrieveRight());

printf("Retrieved from right: %c\n", retrieveRight());

printf("Retrieved from left: %c\n", retrieveLeft());

printf("Retrieved from left: %c\n", retrieveLeft());

return 0;

}

**Question 88 : Write a program to pick up the largest number from a 5 row by 5 column matrix.**

Sol.

# include<stdio.h>

int main( )

{

int a[ 5 ][ 5 ] = { { 11, 1, 7, 9, 7 }, { 13, 54, 56, 2, 5 }, { 23, 43, 89, 22, 13 }, { 14, 15, 17, 16, 19 }, { 45, 3, 6, 8, 10 } } ;

int i, j, big ;

big = a[ 0 ][ 0 ] ;

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

{

for ( j = 0 ; j <= 4 ; j++ )

{

if ( a[ i ][ j ] > big )

big = a[ i ][ j ] ;

}

}

printf ( "\nLargest number in the matrix is %d\n", big ) ;

return 0 ;

}

**Question 89 : Write a program to obtain transpose of a 3 x 5 matrix. The transpose of a matrix is obtained by exchanging the elements of each row with the elements of the corresponding column**

Sol.

#include<stdio.h>

int main( )

{

int mat1[ 3 ][ 5 ] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 } ;

int mat2[ 5 ][ 3 ] ;

int i, j ;

for ( i = 0 ; i < 3 ; i++ )

{

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

mat2[ j ][ i ] = mat1[ i ][ j ] ;

}

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

{

for ( j = 0 ; j < 3 ; j++ )

printf ( "%d\t", mat2[ i ][ j ] ) ;

printf ( "\n" ) ;

}

return 0 ;

}

**Question 90 : [A] What will be the output of the following programs?**

1. **# include<stdio.h>**

**int main( )**

**{**

**int n[ 3 ][ 3 ] = { { 2, 4, 3 }, { 6, 8, 5 }, { 3, 5, 1 } } ;**

**printf ( "%d %d %d\n", \*n, n[ 1 ][ 1 ], n[ 2 ][ 2 ] ) ;**

**return 0 ;**

**}**

1. **# include<stdio.h>**

**int main( )**

**{**

**int n[ 3 ][ 3 ] = { { 2, 4, 3 }, { 6, 8, 5 }, { 3, 5, 1 } } ;**

**int i, \*ptr ;**

**ptr = &n[ 0 ][ 0 ] ;**

**for ( i = 0 ; i <= 8 ; i++ )**

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

**return 0 ;**

**}**

1. **# include<stdio.h>**

**int main( )**

**{**

**int n[ 3 ][ 3 ] = { 2, 4, 3, 6, 8, 5, 3, 5, 1 } ;**

**int i, j ;**

**for ( i = 0 ; i <= 2 ; i++ )**

**for ( j = 0 ; j <= 2 ; j++ )**

**printf ( "%d %d\n", n[ i ][ j ], \*( \*( n + i ) + j ) ) ;**

**return 0 ;**

**}**

Sol.

(A)

2 8 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

**Question 91 : [B] Point out the errors, if any, in the following programs:**

1. **# include<stdio.h>**

**int main( )**

**{**

**int twod[ ][ ] = { 2, 4, 6, 8 } ;**

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

**return 0 ;**

**}**

1. **# include<stdio.h>**

**int main( )**

**{**

**int three[ 3 ][ ] = { { 2, 4, 3 }, { 6, 8, 2 }, { 2, 3, 1 } } ;**

**printf ( "%d\n", three[ 1 ][ 1 ] ) ;**

**return 0 ;**

**}**

Sol.

(A)

#include <stdio.h>

int main() {

int twod[][2] = { {2, 4}, {6, 8} };

printf("%p\n", (void\*)twod);

return 0;

}

(B)

#include <stdio.h>

int main() {

int three[3][3] = { {2, 4, 3}, {6, 8, 2}, {2, 3, 1} };

printf("%d\n", three[1][1]);

return 0;

}

**Question 92 : [C] Attempt the following questions:**

**(a) Match the following with reference to the program segment given below:**

**int i, j, = 25 ;**

**int \*pi, \*pj = & j ;**

**\*pj = j + 5 ;**

**j = \*pj + 5 ;**

**pj = pj ;**

**\*pi = i + j ;**

**Each integer quantity occupies 2 bytes of memory. The value assigned to i begins at (hexadecimal) address F9C and the value assigned to j begins at address F9E. Match the value represented by expression in left column with values in the right column.**

**1. &i a. 30**

**2. &j b. F9E**

**3. pj c. 35**

**4. \*pj d. FA2**

**5. i e. F9C**

**6. pi f. 67**

**7. \*pi g. unspecified**

**8. ( pi + 2 ) h. 65**

**9. (\*pi + 2) i. F9E**

**10. \* ( pi + 2 ) j. F9E**

**k. FAO**

**l. F9D**

Sol.

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

**Question 93 : (A) Match the following with reference to the following program segment:**

**int x[ 3 ][ 5 ] = {**

**{ 1, 2, 3, 4, 5 },**

**{ 6, 7, 8, 9, 10 },**

**{ 11, 12, 13, 14, 15 }**

**}, \*n = &x ;**

**1. \*( \*( x + 2 ) + 1) a. 9**

**2. \*( \*x + 2 ) + 5 b. 13**

**3. \*( \*( x + 1) ) c. 4**

**4. \*( \*( x ) + 2 ) + 1 d. 3**

**5. \* ( \*( x + 1 ) + 3 ) e. 2**

**6. \*n f. 12**

**7. \*( n +2 ) g. 14**

**8. (\*(n + 3 ) + 1 h. 7**

**9. \*(n + 5)+1 i. 1**

**10. ++\*n j. 8**

**k. 5**

**l. 10**

**m. 6**

**(B) Match the following with reference to the following program segment:**

**unsigned int arr[ 3 ][ 3 ] = {**

**{ 2, 4, 6 }, { 9, 1, 10 }, { 16, 64, 5 }**

**} ;**

**1. \*\*arr a. 64**

**2. \*\*arr < \*( \*arr + 2 ) b. 18**

**3. \*( arr + 2 ) / ( \*( \*arr + 1 ) > \*\*arr ) c. 6**

**4. \*( arr[ 1 ] + 1 ) | arr[ 1 ][ 2 ] d. 3**

**5. \*( arr[ 0 ] ) | \*( arr[ 2 ] ) e. 0**

**6. arr[ 1 ][ 1 ] < arr[ 0 ][ 1 ] f. 16**

**7. arr[ 2 ][ [ 1 ] & arr[ 2 ][ 0 ] g. 1**

**8. arr[ 2 ][ 2 ] | arr[ 0 ][ 1 ] h. 11**

**9. arr[ 0 ][ 1 ] ^ arr[ 0 ][ 2 ] i. 20**

**10. ++\*\*arr + --arr[ 1 ][ 1 ] j. 2**

**k. 5**

**l. 4**

Sol.

(A)

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

(B)

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

**Question 94 : (a) Write a program to find if a square matrix is symmetric.**

**(b) Write a program to add two 6 x 6 matrices.**

**(c) Write a program to multiply any two 3 x 3 matrices.**

**(d) Given an array p[ 5 ], write a function to shift it circularly left by two positions. Thus, if the original array is { 15, 30, 28, 19, 61 } then after shifting it will be { 28, 19, 61, 15, 30 } Call this function for a 4 x 5 matrix and get its rows left shifted.**

Sol.

(A)

#include<stdio.h>

int main()

{

int i,j,count=0;

printf("Enter no. of rows & coulumns = ");

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;

}

(B)

#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;

}

(C)

#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");

}

}

(D)

#include<stdio.h>

#include<conio.h>

void shift(int \*base)

{

int \*web, fir, sec, i;

web = base;

fir = \*base;

sec = \*(base + 1);

for (i = 0; i<3; i++)

\*(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++)

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

shift(a);

printf("\n\nList after shifting it's rows by two positions.\n\n");

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

printf("%d ", a[i]);

\_getch();

return 0;

}

**Question 95 : Write a program that extracts part of the given string from the specified position. For example, if from the sting "Working with strings is fun", starting from position 3, 4 characters are extracted then it should return "king".**

Sol.

# include<stdio.h>

# include<stdlib.h>

# include<string.h>

int main( )

{

char str[ 20 ], news[ 20 ] ;

char \*s, \*t ;

int pos, n, i ;

printf ( "\nEnter a string: " ) ;

scanf ( "%s", str ) ;

printf ( "Enter position and no. of characters to extract: " ) ;

scanf ( "%d %d", &pos, &n ) ;

s = str ; t = news ;

if ( pos < 0 || pos > strlen ( str ) )

{

printf ( "Improper position value" ) ;

exit ( 1 ) ;

}

if ( n < 0 )

n = 0 ;

if ( n > strlen ( str ) )

n = n - strlen ( str ) - 1 ;

s = s + pos ;

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

{

\*t = \*s ; s++ ; t++ ;

}

\*t = '\0' ;

printf ( "The substring is: %s\n", news ) ;

return 0 ;

}

**Question 96 : Write a program that converts a string like "124" to an integer 124.**

Sol.

# include<stdio.h>

int main( )

{

char str[ 6 ] ;

int num = 0, i ;

printf ( "Enter a string containing a number: " ) ;

scanf ( "%s", str ) ;

for ( i = 0 ; str [ i ] != '\0' ; i++ )

{

if ( str[ i ] >= 48 && str[ i ] <= 57 )

num = num \* 10 + ( str[ i ] - 48 ) ;

else

{

printf ( "Not a valid string\n" ) ;

return 1 ;

}

}

printf ( "The number is: %d\n", num ) ;

return 0 ;

}

**Question 97 : Write a program that generates and prints the Fibonacci words of order 0 through 5. For example, f(0) = "A", f(1) = "B", f(2) = "BA", f(3) = "BAB", f(4) = "BABBA", etc.**

Sol.

#include<stdio.h>

#include<string.h>

int main( )

{

char str[ 50 ] ;

char lastbutoneterm[ 50 ] = "A" ;

char lastterm[ 50 ] = "B" ;

int i ;

for ( i = 1 ; i <= 5 ; i++ )

{

strcpy ( str, lastterm ) ;

strcat ( str, lastbutoneterm ) ;

printf ( "%s\n", str ) ;

strcpy ( lastbutoneterm, lastterm );

strcpy ( lastterm, str ) ;

}

return 0 ;

}

**Question 98 : [A] What will be the output of the following programs?**

1. **# include<stdio.h>**

**int main( )**

**{**

**char c[ 2 ] = "A" ;**

**printf ( "%c\n", c[ 0 ] ) ;**

**printf ( "%s\n", c ) ;**

**return 0 ;**

**}**

1. **# include<stdio.h>**

**int main( )**

**{**

**char s[ ] = "Get organized! Learn C!!" ;**

**printf ( "%s\n", &s[ 2 ] ) ;**

**printf ( "%s\n", s ) ;**

**printf ( "%s\n", &s ) ;**

**printf ( "%c\n", s[ 2 ] ) ;**

**return 0 ;**

**}**

1. **# include<stdio.h>**

**int main( )**

**{**

**char s[ ] = "Borrowers of books spoil the symmetry of shelves" ;**

**int i = 0 ;**

**while ( s[ i ] != 0 )**

**{**

**printf ( "%c %c\n", s[ i ], \*( s + i ) ) ;**

**printf ( "%c %c\n", i[ s ], \*( i + s ) ) ;**

**i++ ;**

**}**

**return 0 ;**

**}**

1. **# include<stdio.h>**

**int main( )**

**{**

**char str1[ ] = { ’H’, ’e’, ’l’, ’l’, ’o’, 0 } ;**

**char str2[ ] = "Hello" ;**

**printf ( "%s\n", str1 ) ;**

**printf ( "%s\n", str2 ) ;**

**return 0 ;**

**}**

1. **# include<stdio.h>**

**int main( )**

**{**

**printf ( 5 + "Good Morning " ) ;**

**printf ( "%c\n", "abcdefgh"[ 4 ] ) ;**

**return 0 ;**

**}**

1. **# include<stdio.h>**

**int main( )**

**{**

**printf ( "%d %d %d\n", sizeof ( ’3’ ), sizeof ( "3" ), sizeof ( 3 ) ) ;**

**return 0 ;**

**}**

Sol.

1. A

A

1. t organized! Learn C!!

Get organized! Learn C!!

Get organized! Learn C!!

t

1. B B

B B

o o

o o

r r

r r

r r

r r

o o

o o

w w

w w

e e

e e

r r

r r

s s

s s

' '

' '

o o

o o

f f

f f

' '

' '

b b

b b

o o

o o

o o

o o

k k

k k

s s

s s

' '

' '

s s

s s

p p

p p

o o

o o

i i

i i

l l

l l

' '

' '

t t

t t

h h

h h

e e

e e

' '

' '

s s

s s

y y

y y

m m

m m

m m

m m

e e

e e

t t

t t

r r

r r

y y

y y

' '

' '

o o

o o

f f

f f

' '

' '

s s

s s

h h

h h

e e

e e

l l

l l

v v

v v

e e

e e

s s

s s

1. Hello

Hello

1. Morning

e

1. 4 2 4

**Question 99 : [B] Fill in the blanks:**

**(a) "A" is a \_\_\_\_\_\_\_\_\_\_\_ whereas ’A’ is a \_\_\_\_\_\_\_\_\_\_\_\_.**

**(b) A string is terminated by a \_\_\_\_\_\_ character.**

**(c) The array char name[ 10 ] can consist of a maximum of \_\_\_\_\_\_ characters.**

**(d) The array elements are always stored in \_\_\_\_\_\_\_\_\_ memory locations.**

Sol.

1. "A" is a **string** whereas 'A' is a **character**.
2. A string is terminated by a **null** character.
3. The array char name[10] can consist of a maximum of **9** characters.
4. The array elements are always stored in **contiguous** memory locations.

**Question 100 : [C] Attempt the following questions:**

**(a) If the string "Alice in wonder land" is fed to the following scanf( ) statement, what will be the contents of arrays str1, str2, str3 and str4? scanf ( "%s%s%s%s", str1, str2, str3, str4 ) ;**

**(b) To uniquely identify a book a 10-digit ISBN (International Standard Book Number) is used. The rightmost digit in ISBN is a checksum digit. This digit is determined from the other 9 digits using the condition that d1 + 2d2 + 3d3 + ... + 10d10 must be a multiple of 11 (where di denotes the ith digit from the right). The checksum digit d1 can be any value from 0 to 10: the ISBN convention is to use the value X to denote 10. Write a program that receives a 10-digit integer, computes the checksum, and reports whether the ISBN number is correct or not.**

**(c) A Credit Card number is usually a 16-digit number. A valid Credit Card number would satisfy a rule explained below with the help of a dummy Credit Card number—4567 1234 5678 9129. Start with the rightmost - 1 digit and multiply every other digit by 2.**

**4 5 6 7 1 2 3 4 5 6 7 8 9 1 2 9**

**8 12 2 6 10 14 18 4**

Sol.

a>

str1 : Alice

str2 : in

str3 : wonder

str4 : land

b>

#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;

sum = sum + ((i + 1)\*isbn[i]);

}

if (sum % 11)/\*If not divisble by 11\*/

puts("\nISBN number is wrong.");

else

puts("\nISBN number is valid.");

getch();

return 0;

}

c>

#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++)

{

num[i] -= 48;

if ((i % 2))

sum = sum + num[i];

else

{

num[i] \*= 2;

if (num[i] >= 10)

num[i] -= 9;

sum = sum + num[i];

}

}

if (!(sum % 10))

printf("\nNumber is valid.");

else

printf("\nNumber is not valid.");

getch();

return 0;

}