**KPIT TECHNOLOGIES**

**WEEKLY REPORT**

**WEEK 1- Report (DATE: 14/06/2024)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Student name** | **Week** | **Branch** | **USN** |
| **C Tejeswar** | **4** | **Circuit (ECE)** | **1NH20EC028** |

**Yashavant Kanetkar Book**

**151-200:**

Q151. What will be the output of the following programs?

# include <stdio.h>

void main()

{

enum status { pass, fail, atkt } ;

enum status stud1, stud2, stud3 ;

stud1 = pass ;

stud2 = fail ;

stud3 = atkt ;

printf ( "%d %d %d\n", stud1, stud2, stud3 ) ;

}

Output:

0 1 2

Q152. # include <stdio.h>

void main()

{

printf ( "%f\n", ( float ) ( ( int ) 3.5 / 2 ) ) ;

printf ( "%d\n", ( int ) ( ( ( float ) 3 / 2 ) \* 3 ) ) ;

}

Output:

1.000000

4

Q153. Point out the error, if any, in the following programs:

# include <stdio.h>

void main()

{

typedef struct patient

{

char name[ 20 ] ; int age ;

int systolic\_bp ; int diastolic\_bp ;

} ptt ;

ptt p1 = { "anil", 23, 110, 220 } ;

printf ( "%s %d\n", p1.name, p1.age ) ;

printf ( "%d %d\n", p1.systolic\_bp, p1.diastolic\_bp ) ;

}

No Error

Q154. # include <stdio.h>

void show( ) ;

void main()

{

void ( \*s )( ) ;

s = show ;

( \*s )( ) ;

s( ) ;

}

void show( )

{

printf ( "don't show off. It won't pay in the long run\n" ) ;

}

No Error

Q155. # include <stdio.h>

void show ( int, float ) ;

void main()

{

void ( \*s )( int, float ) ;

s = show ;

( \*s )( 10, 3.14) ;

}

void show ( int i, float f )

{

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

}

No Error

Q156.Write a program, which stores information about a date in a structure containing three members—day, month and year. Using bit fields, the day number should get stored in first 5 bits of day, the month number in 4 bits of month and year in 12 bits of year. Write a program to read date of joining of 10 employees and display them in ascending order of year.

Sol.

# include <stdio.h>

void main()

{

struct date

{

unsigned day : 5 ;

unsigned month : 4 ;

unsigned year : 12 ;

} ;

struct date dt[ 10 ], temp ;

int i, j, d, m, y ;

printf ( "Enter joining dates (dd-mm-yyyy) of 10 employees\n" ) ;

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

{

scanf ( "%d %d %d", &d, &m, &y ) ;

if ( d < 1 || d > 31 || m < 1 || m > 12 )

{

printf ( "Invalid date, enter new date\n" ) ;

i-- ;

continue ;

}

dt[ i ].day = d ;

dt[ i ].month = m ;

dt[ i ].year = y ;

}

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

{

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

{

if ( dt[ j ].year < dt[ i ].year )

{

temp = dt[ i ] ;

dt[ i ] = dt[ j ] ;

dt[ j ] = temp ;

}

}

}

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

printf ( "%d %d %d\n", dt[ i ].day, dt[ i ].month, dt[ i ].year ) ;

}

Q157. Write a program to read and store information about insurance policy holder. The information contains details like gender, whether the holder is minor/major, policy name and duration of the policy. Make use of bit-fields to store this information.

Sol.

#include <stdio.h>

# include <string.h>

void main()

{

struct policy\_holder

{

unsigned gender : 1 ; // 0-Male, 1-Female

unsigned status : 1 ; // 0-Minor, 1-Major

char name[ 20 ] ;

unsigned dr : 5 ;

} ;

struct policy\_holder h ;

int g, s, d ;

char n[ 20 ] ; printf ( "\nEnter gender (0-Male, 1-Female): " ) ;

scanf ( "%d", &g ) ;

printf ( "\nEnter status (0-Minor, 1-Major): " ) ;

scanf ( "%d", &s ) ;

printf ( "\nEnter name of the policy holder: " ) ;

scanf ( "%s", n ) ;

printf ( "\nEnter duration (1 to 25 yrs) of the policy: " ) ;

scanf ( "%d", &d ) ;

h.gender = g ;

h.status = s ;

strcpy ( h.name, n ) ;

h.dr = d ;

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

printf ( "Gender: %s\n", h.gender == 0 ? "Male" : "Female" ) ;

printf ( "Status: %s\n", h.status == 0 ? "Minor" : "Major" ) ;

printf ( "Duration %d\n", h.dr ) ;

}

Q158.Write a program to calculate the sum of the following series:

1! 2! + 2! 3! + 3! 4! + 4! 5! + …… + 9! 10!

#include <stdio.h>

void main()

{

int i, j ;

float prod1, prod2, term, s ;

s = 0 ;

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

{

prod1 = 1 ;

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

prod1 = prod1 \* j ;

prod2 = prod1 \* j ;

term = prod1 \* prod2 ;

s = s + term ;

}

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

}

Q159. Write a program to enter the numbers till the user wants and at the end it should display the count of positive, negative and zeros entered.

#include <stdio.h>

void main()

{

int neg, pos, zero, n ;

char ch = 'y' ;

pos = neg = zero = 0 ;

while ( ch == 'y' || ch == 'Y' )

{

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

scanf ( "%d", &n ) ;

if ( n > 0 )

pos++ ;

if ( n < 0 )

neg++ ;

if ( n == 0 )

zero++ ;

printf ( "Do you want to continue y/n" ) ;

fflush ( stdin ) ;

scanf ( "%c", &ch ) ;

}

printf ( "Positive = %d\n", pos ) ;

printf ( "Negative = %d\n", neg ) ;

printf ( "Zeros = %d\n", zero ) ;

}

Q160. Write a program to find the range of a set of numbers that are input through the keyboard. Range is the difference between the smallest and biggest number in the list.

#include<stdio.h>

void main()

{

int n, no, flag, small, big ;

flag = 0 ;

printf ( "Enter the number of elements in the range:\n" ) ;

scanf ( "%d", &n ) ;

while ( n > 0 )

{

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

scanf ( "%d", &no ) ;

if ( flag == 0 )

{

small = big = no ;

flag = 1 ;

}

else

{

if ( no > big )

big = no ;

if ( no < small )

small = no ;

}

n-- ;

}

printf ( "Range: %d", big - small ) ;

}

Q161. If three integers are entered through the keyboard, write a program to determine whether they form a Pythagorean triplet or not.

#include<stdio.h>

void main()

{

int i, j, k ;

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

scanf ( "%d%d%d", &i, &j, &k ) ;

if ( ( i \* i + j \* j == k \* k ) || ( j \* j + k \* k == i \* i ) ||

( k \* k + i \* i == j \* j ) )

printf ( "Numbers form pythogorean triplet \n" ) ;

else

printf ( "Numbers do not form pythogorean triplet \n" ) ;

}

Q162.

A recursive call should always be subjected to an **if**. Why? Explain with an example.

If the recursive call is not subjected to an **if**, the function would fall in an infinite loop.

void fun( )

{

static int count = 0 ;

count++ ;

if ( count <= 5 )

{

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

fun( ) ;

}

Else

return ;

}

Q163.

Define a function that receives 4 integers and returns sum, product and average of these integers.

#include <stdio.h>

void calc ( int \*, int \*, float \* ) ;

void main()

{

int sum, prod ;

float avg ;

calc ( &sum, &prod, &avg ) ;

printf ( "%d %d %f\n", sum, prod, avg ) ;

}

void calc ( int \*sum, int \*prod, float \*avg )

{

int n1, n2, n3, n4 ;

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

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

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

\*prod = n1 \* n2 \* n3 \* n4 ;

\*avg = \*sum / 4.0f ;

}

Q163. Define a recursive function which prints the prime factors of the number that it receives when called from **main( )** .

#include <stdio.h>

void factor ( int ) ;

void main()

{

int num ;

printf ( "\nEnter a number: " ) ;

scanf ( "%d", &num ) ;

printf ( "\nPrime Factors are: " ) ;

factor ( num ) ;

}

void factor ( int n )

{

static int i = 2 ;

if ( i <= n )

{

if ( n % i == 0 )

{

printf ( "%d ", i ) ;

n = n / i ;

}

else

i++ ;

factor ( n ) ;

}

return ;

}

Q164. Define macros for calculating area of circle, circumference of circle, volume of a cone and volume of sphere.

#include <stdio.h>

#define PI 3.14f

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

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

#define VCO( r ) ( ( 1.0f / 3 ) \* PI \* r \* r \* r )

#define VCYL( r ) ( ( 4.0f / 3 ) \* PI \* r \* r \* r )

void main()

{

float r, a\_ci, c\_ci, v\_co, v\_cyl ;

printf ( "\n Enter the value of r :\n" ) ;

scanf ( "%f", &r ) ;

a\_ci = ACI ( r ) ;

c\_ci = CCI ( r ) ;

printf ( "\nArea of circle : %f", a\_ci ) ;

printf ( "\nCircumference of circle : %f", c\_ci ) ;

v\_co = VCO ( r ) ;

v\_cyl = VCYL ( r ) ;

printf ( "\n Volume of cone : %f",v\_co ) ;

printf ( "\n Volume of cylinder : %f", v\_cyl ) ;

}

Q165. Write a program that prints sizes of all types of chars, ints and reals.

#include <stdio.h>

void main()

{

char ch ;

unsigned char dh ;

printf ( "character = %d\n", sizeof ( ch ) ) ;

printf ( "unsigned character = %d\n", sizeof ( dh ) ) ;

short int a ;

short unsigned int b ;

int c ;

long int d ;

long unsigned int e ;

printf ( "short signed integer = %d\n", sizeof ( a ) ) ;

printf ( "short unsigned integer = %d\n", sizeof ( b ) ) ;

printf ( "integer = %d\n", sizeof ( c ) ) ;

printf ( "long signed integer = %d\n", sizeof ( d ) ) ;

printf ( "long unsigned integer = %d\n", sizeof ( e ) ) ;

float f ;

double g ;

long double h ;

printf ( "float = %d\n", sizeof ( f ) ) ;

printf ( "double = %d\n", sizeof ( g ) ) ;

printf ( "long double = %d\n", sizeof ( h ) ) ;

}

Q166.How will you allocat space for a **float** array whose size is received from the keyboard?

# include <stdlib.h>

# include <stdio.h>

void main()

{

int sz ;

float \*fptr ;

printf ( "Enter size of the array: " ) ;

scanf ( "%d", &sz ) ;

fptr = ( float \* ) malloc ( sz \* sizeof ( float ) ) ;

}

# include <stdio.h>

void main()

{

int sz ;

printf ( "Enter size of the array: " ) ;

scanf ( "%d", &sz ) ;

float arr[ sz ] ;

}

Q167.Write a function that receives as parameters, a 1-D array, its size and an integer and returns number of times the integer occurs in the array.

#include <stdio.h>

int countnum ( int\*, int, int ) ;

void main()

{

int arr[ 100 ], size, num, i, count ;

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

scanf ( "%d", &size ) ;

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

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

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

printf ( "Enter the number you want to count:\n" ) ;

scanf ( "%d", &num ) ;

count = countnum ( arr, size, num ) ;

printf ( "count = %d", count ) ;

}

int countnum ( int \*a, int sz, int n )

{

int j, cnt = 0 ;

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

{

if ( \*a == n )

cnt++ ;

a++ ;

}

return cnt ;

}

Q168. Create an array of pointers containing names of 10 cities. Write a program that sorts the cities in reverse alphabetical order and prints this reversed list.

#include <stdio.h>

#include <string.h>

void main()

{

char \*cities[ ] = {

"Nagpur", "Kanpur", "Delhi",

"Sikandarabad", "Akola", "Ghatanji",

"Jabalpur", "Ziri", "Shegaon", "Bombay"

} ;

char \*t ;

int i, j ;

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

{

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

{

if ( strcmp ( cities[ i ], cities[ j ] ) < 0 )

{

t = cities[ i ] ;

cities[ i ] = cities[ j ] ;

cities[ j ] = t ;

}

}

}

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

printf ( "%s\n", cities[ i ] ) ;

}

Q169. Declare a structure called **student** containing his name, age and address. Create and initialize three structure variables. Define a function to which these variables are passed. The function should

convert the names into uppercase. Print the resultant structure variables.

#include <stdio.h>

#include <string.h>

void upper ( struct stud\* ) ;

struct stud

{

char name[ 20 ] ;

int age ;

char addr[ 40 ] ;

} ;

struct stud s1 = { "akshay", 20, "Ravinagar" } ;

struct stud s2 = { "shubham", 21, "Civil Lines" } ;

struct stud s3 = { "nilesh", 22, "Khamla" } ;

void main()

{

upper ( &s1 ) ;

upper ( &s2 ) ;

upper ( &s3 ) ;

}

void upper ( struct stud \*s )

{

printf ( "Before conversion:\n" ) ;

printf ( "%s %d %s\n", s->name, s->age, s->addr ) ;

strupr ( s->name ) ;

printf ( "After conversion:\n" ) ;

printf ( "%s %d %s\n", s->name, s->age, s->addr ) ;

}

Q170. Write a program that checks and reports whether sum of elements in the ith row of a 5 x 5 array is equal to sum of elements in ith column.

#include <stdio.h>

void main()

{

int a[ 5 ][ 5 ], i, j, sumr = 0, sumc = 0 ;

printf ( "Enter elements of a 5 x 5 array\n" ) ;

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

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

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

printf ( "Enter row and column you wish to check\n" ) ;

scanf ( "%d", &i ) ;

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

sumr = sumr + a[ i - 1 ][ j ] ;

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

sumc = sumc + a[ j ][ i - 1 ] ;

if ( sumr == sumc )

printf ( "Sums are equal" ) ;

else

printf ( "Sums are not equal" ) ;

}

Q171.

Write a program to multiply two integers using bitwise operators.

#include <stdio.h>

int add ( int, int ) ;

void main()

{

int a, b, result ;

printf ( "\nEnter the numbers to be multiplied :" ) ;

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

result = 0 ;

while ( b != 0 )

{

if (b & 1 )

result = add ( result, a ) ;

a <<= 1 ;

b >>= 1 ;

}

printf ( "Result:%d", result ) ;

}

int add ( int x, int y )

{

while ( y != 0 )

{

int carry = x & y ;

x = x ^ y ;

y = carry << 1 ;

}

return x ;

}

Q172.Write a program to count number of words in a given text file.

#include <stdio.h>

#include <stdlib.h>

void main()

{

char ch ; FILE \*fp ;

char fname[ 67 ] ;

int count = 0 ;

printf ( "Enter File name: " ) ;

gets ( fname ) ;

fp = fopen ( fname, "r" ) ;

if ( fp == NULL )

{

printf ("Unable to open file\n" ) ;

exit ( 1 ) ;

}

while ( ( ch = getc ( fp ) ) != EOF )

{

if ( ch == ' ' )

count++ ;

}

printf ( "No of words = %d", count + 1 ) ;

fclose ( fp ) ;

}

Q173. Write a program that receives a set of numbers as command- line arguments and prints their average.

#include <stdio.h>

#include <stdlib.h>

int main ( int argc, char\* argv[ ] )

{

int sum, i, avg ;

sum = 0 ;

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

sum = sum + atoi ( argv[ i ] ) ;

avg = sum / ( argc - 1 ) ;

printf ( "Average = %d", avg ) ;

}

Q175. Write a program to check whether contents of the two files are same by comparing them on a byte-by-byte basis.

#include <stdio.h>

#include <stdlib.h>

void main()

{

char ch1,ch2 ;

FILE \*fp,\*fq ;

char fname1[ 67 ], fname2[ 67 ] ;

printf ( "Enter File1 name: " ) ;

gets ( fname1 ) ;

printf ( "Enter File2 name: " ) ;

gets ( fname2 ) ;

fp = fopen ( fname1, "rb" ) ;

if ( fp == NULL )

{

printf ("Unable to open file1\n" ) ;

exit ( 1 ) ;

}

fq = fopen ( fname2, "rb" ) ;

if ( fq == NULL )

{

printf ("Unable to open file2\n" ) ;

exit ( 1 ) ;

}

while ( 1 )

{

ch1 = getc ( fp ) ;

ch2 = getc ( fq ) ;

if ( ch1 == EOF && ch2 == EOF )

{

printf ( "File contents match" ) ;

break ;

}

if ( ch1 != ch2 || ch1 == EOF && ch2 != EOF ||

ch1 != EOF && ch2 == EOF )

{

printf ( "File contents do not match" ) ;

break ;

}

}

fclose ( fp ) ;

fclose ( fq ) ;

}

Q177. Dynamically allocate space for a 3-D array of dimensions 3 x 5 x 4. Set up each elements of this array with a value 10. Report an error, if enough memory space is not available.

#include <stdio.h>

#include <stdlib.h>

void main()

{

int i, j, k ;

int \*a ;

a = ( int \* ) malloc ( 3 \* 5 \* 4 \* sizeof ( int ) ) ;

if ( a == NULL )

printf ( "Insufficient space\n" ) ;

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

{

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

{

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

{

\*( a + i \* 5 \* 4 + j \* 4 + k ) = 10 ;

printf ( "%d ", \*( a + i \* 5 \* 4 + j \* 4 + k ) = 10 ) ;

}

printf ( "\n" ) ;

}

printf ( "\n\n" ) ;

}

}

Q178.Point out errors, in any, in the following code snippet:

int a ;

float b ;

char ch ;

scanf ( "%d %f %c", a, b, ch ) ;

printf ( "%d %f %c", &a, &b, &ch ) ;

Error. “&” should be used before each variable in **scanf( )**.

Q179.Write a program that defines a function called **isalnum( )**. The function should receive a string and check if all characters in it are alphabets or digits. If so, it should return a true, otherwise false. Call this function for the following strings:

"ABCD1234"

"Nagpur – 440010"

#include <stdio.h>

#define TRUE 1

#define FALSE 0

int isalnum ( char \* ) ;

void main()

{

char str1[ ] = "ABCD1234" ;

char str2[ ] = "Nagpur – 440010" ;

int ret ;

ret = isalnum ( str1 ) ;

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

ret = isalnum ( str2 ) ;

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

}

int isalnum ( char \*p )

{

while ( \*p )

{

if ( \*p >= '0' && \*p <= '9' || \*p >= 'a' && \*p <= 'z' ||

\*p >= 'A' && \*p <= 'Z' )

p++ ;

else

return FALSE ;

}

return TRUE ;

}

Q180.Define an enumeration to represent colors red, green, yellow, magenta and brown. Create two variables **Apple** and **Banana** of this **enum** type and assign colors red and yellow to them respectively. Print these color values and indicate what output will they produce.

#include <stdio.h>

#include <stdlib.h>

void main()

{

enum fruits { red, green, yellow, magenta, brown } ;

enum fruits Apple, Banana ;

Apple = red ;

Banana = yellow ;

printf ( "%d %d\n", Apple, Banana ) ;

}

Output

0 2

Q181. Define a function called **showbits( )** which displays all the bits of an **unsigned char** that it receives. Call this function for values 45 and 30. Indicate what output will **showbits( )** produce for these values?

# include <stdio.h>

void showbits ( unsigned char ) ;

void main()

{

unsigned char num1 = 45, num2 = 30 ;

showbits ( num1 ) ;

showbits ( num2 ) ;

}

void showbits ( unsigned char n )

{

int i ;

unsigned char j, k, andmask ;

for ( i = 7 ; i >= 0 ; i-- )

{

j = i ;

andmask = 1 << j ;

k = n & andmask ;

k == 0 ? printf ( "0" ) : printf ( "1" ) ;

}

printf ( "\n" ) ;

}

Q182. Write a program to generate and print all unique combinations of numbers 1, 2, 3 and 4.

#include <stdio.h>

void main()

{

int i, j, k, l ;

for ( i = 1 ; i <= 4 ; i++ )

{

for ( j = 1 ; j <= 4 ; j++ )

{

for ( k = 1 ; k <= 4 ; k++ )

{

for ( l = 1 ; l <= 4 ; l++ )

{

if ( i != j && i != k && i != l && j != k && j != l

&& k != l )

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

}

}

}

}

}

Q184. Define an iterative function and a recursive function to print first ten terms of a Fibonacci series. Which of these two functions will run faster and why?

# include <stdio.h>

void ifibo ( int, int, int ) ;

void rfibo ( int, int, int, int ) ;

void main()

{

int first = 0, second = 1 ;

printf ( "First 10 terms of Fibonacci series are:\n" ) ;

ifibo ( first, second, 10 ) ;

printf ( "\n" ) ;

rfibo ( first, second, 0, 10 ) ;

}

void ifibo ( int old, int current, int terms )

{

int newterm, i ;

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

{

if ( i <= 1 )

newterm = i ;

else

{

newterm = old + current ;

old = current ;

current = newterm ;

}

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

}

}

void rfibo ( int old, int current, int count, int terms )

{

int newterm ;

if ( count == 0 || count == 1 )

{

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

count++ ;

terms-- ;

rfibo ( old, current, count, terms ) ;

}

else if ( terms >= 1 )

{

newterm = old + current ;

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

terms = terms - 1 ;

old = current ;

current = newterm ;

rfibo ( old, current, count, terms ) ;

}

}

Q185.Create a data structure for storing the following data:

Name of document – Leaflet / Flier / Brochure

Number of colors – 1 / 2 / 4 / 5

Size – Small / Medium / Big

Type of paper – Maplitho / Bond / Artcard

Number of copies

Type of printing – Positive / Negative

struct doc

{

char name[ 20 ] ;

int numcolors ;

int size ;

char papertype ;

int numcopies ;

} ;

struct doc d1, d2, d3 ;

Q187.Consider the following structure:

struct Flower

{

char name[ 20 ] ;

int color ;

int no\_of\_petals ;

} ;

struct Flower f[ 3 ] ;

Write statements to receive values into array **f[ ]** and print them on the screen.

#include <stdio.h>

#include <stdlib.h>

struct Flower

{

char name[ 20 ] ;

int color ;

int no\_of\_petals ;

} ;

struct Flower f[ 3 ] ;

void main()

{

int i ;

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

{

printf( "Enter %d flower name, color and no of petals: ", i ) ;

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

scanf( "%d", &f[ i ].color ) ;

scanf( "%d", &f[ i ].no\_of\_petals ) ;

}

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

printf("%s %d %d\n" ,f[i].name, f[i].color, f[i].no\_of\_petals) ;

}

Q188. Given two matrices **A3 X 3** and **B3 X 3**, define a function that checks whether matrix A is transpose of matrix B.

#include <stdio.h>

#define N 3

int istranspose ( int B[ ][N], int A[ ][N] ) ;

void main()

{

int B[ N ][ N ] = { { 1, 2, 1 }, { 2, 1, 2 }, { 1, 2, 1 } } ;

int A[ N ][ N ] = { { 1, 2, 1 }, { 2, 1, 2 }, { 1, 2, 1 } } ;

int flag ;

flag = istranspose ( B, A ) ;

if ( flag == 1 )

printf ( "Matrix A is transpose of Matrix B\n" ) ;

else

printf ( "Matrix A is not a transpose of Matrix B\n" ) ;

}

int istranspose ( int B[ ][ N ], int A[ ][ N ] )

{

int i, j, flag ;

flag = 1 ;

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

{

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

{

if ( B[ i ][ j ] != A[ j ][ i ] )

}

}

return 1 ;

}

Q189.Write a code snippet to carry out the following operations:

Open a file ‘records.dat’ in read binary mode.

Skip first 200 bytes from the beginning of the file.

Read next 20 bytes into an array **arr[ ]**.

#include <stdio.h>

#include <stdlib.h>

void main()

{

FILE \*fp ;

char arr[ 25 ] ;

fp = fopen ( "records.dat", "rb" ) ;

{

puts ( "Cannot open file\n" ) ;

exit ( 1 ) ;

}

fseek ( fp, 200L, SEEK\_SET ) ;

fread ( arr, 20, 1, fp ) ;

fclose( fp ) ;

}

Q190. While using command-line arguments, is it necessary to use variables **argc** and **argv**? Write statements to print the name of the executable file and the first and second argument passed to the program?

No. Any other variables other than **argc** and **argv** can be used.

#include <stdio.h>

#include <stdlib.h>

int main ( int ac, char \*av[ ] )

{

printf ( "%s %s %s", av[ 0 ], av[ 1 ], av[ 2 ] ) ;

}

Q191.What will be the output of the following program?

# include <stdio.h>

void main()

{

int i = 32, j = 65, k, l, m, n, o, p ;

k = i | 35 ;

l = ~k ;

m = i & j ;

n = j ^ 32 ;

o = j << 2 ;

p = i >> 5 ;

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

printf ( "n = %d o = %d p = %d\n", n, o, p ) ;

}

*Output:*

k = 35 l = -36 m = 0

n = 97 o = 260 p = 1

Q192.Consider an unsigned integer in which rightmost bit is numbered as 0. Write a function **checkbits ( x, p, n )** which returns true if all "n" bits starting from position "p" are turned on. For example, **checkbits ( x, 4, 3 )** will return true if bits 4, 3 and 2 are 1 in number x.

*Sol.*

# include <stdio.h>

#include <math.h>

# define \_BV(x) ( 1 << x )

void showbits ( unsigned int ) ;

int checkbits ( unsigned int, int, int ) ;

void main()

{

unsigned int x = 0xF0FF ;

int n, p ;

int flag ;

printf ( "Value of x = " ) ;

showbits ( x ) ;

printf ( "\n" ) ;

printf ( "Enter position and number of bits:\n" ) ;

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

flag = checkbits ( x, p, n ) ;

if ( flag == 1 )

printf ( "%d bits from pos %d are turned on\n", n, p ) ;

else

printf ( "%d bits from pos %d are not turned on\n", n, p ) ;

}

void showbits ( unsigned int n )

{

int i ;

unsigned int j, k, andmask ;

for ( i = 15 ; i >= 0 ; i-- )

{

j = i ;

k = n & \_BV( j ) ;

k == 0 ? printf ( "0" ) : printf ( "1" ) ;

}

}

int checkbits ( unsigned int x, int p, int n )

{

int i ;

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

{

if ( x & \_BV( p ) != \_BV( p ) )

p-- ;

}

return 1 ;

}

1. Q195. Write a program to scan an 8-bit number into a variable and check whether its 3rd, 6th and 7th bit is on.

*Sol.*

# include <stdio.h>

# define \_BV(x) ( 1 << x )

void showbits ( unsigned char ) ;

void main()

{

unsigned char a ;

unsigned int num ;

printf ( "\nEnter a number: " ) ;

scanf ( "%hu", &num ) ;

num = num & 0x00FF ;

a = num ;

showbits ( num ) ;

printf ( "\n" ) ;

if ( ( a & \_BV( 3 ) ) == \_BV( 3 ) )

printf ( "Its third bit is on\n" ) ;

else

printf ( "Its third bit is off\n" ) ;

if ( ( a & \_BV( 6 ) ) == \_BV( 6 ) )

printf ( "Its sixth bit is on\n" ) ;

else

printf ( "Its sixth bit is off\n" ) ;

if ( ( a & \_BV( 7 ) ) == \_BV( 7 ) )

printf ( "Its seventh bit is on\n" ) ;

else

printf ( "Its seventh bit is off\n" ) ;

}

void showbits ( unsigned char n )

{

int i ;

unsigned char k, andmask ;

for ( i = 7 ; i >= 0 ; i-- )

{

andmask = 1 << i ;

k = n & andmask ;

k == 0 ? printf ( "0" ) : printf ( "1" ) ;

}

}

Q196.Consider the following statements:

int a = 20 ;

int \*p ;

p = &a ;

Write statements only using p to:

− set a value 45 in a.

− multiply **a** with 40 and store the result in **a**

− print current value of a

\*p = 45;

\*p = \*p \* 40;

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

Also write statements to perform the following operations:

− Increment p

− After incrementation, what will be present in **p** if variable **a** is at location 4004?

− Does incrementing **p** cause a memory leak?

p++ ;

After incrementation p will contain 4008.

No incrementing p does not cause memory leak.

Q197. Write a program that defines a function called **isalpha( )**. The function should receive a string and check if all characters in it are alphabets. If so, it should return a true, otherwise false. Call this function for the following strings:

"NambyPamby"

"Mumbai – 400010"

#include <stdio.h>

#define TRUE 1

#define FALSE 0

int isalpha ( char \* ) ;

void main()

{

char str[ ] = "NambyPamby" ;

int ret ;

ret = isalpha ( str ) ;

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

}

int isalpha ( char \*p )

{

while ( \*p )

{

if ( \*p >= 'a' && \*p <= 'z' || \*p >= 'A' && \*p <= 'Z' )

p++ ;

else

return FALSE ;

}

return TRUE ;

}

Q198.Define an enumeration to represent marital status of a person—single, married, divorced. Create two variables **he** and **she** of this **enum** type and assign to them values single and married respectively. Print these values and indicate what output will they produce.

#include <stdio.h>

#include <stdlib.h>

void main()

{

enum mar\_status { single, married, divorced, widowed } ;

enum mar\_status he, she ;

he = single ;

she = married ;

printf( "%d %d\n",he, she ) ;

}

Output

0 1

Q199. Define functions **countzeros(** ) and **countones( )** which count number of 0s and 1s in an **unsigned char** that they receive. Call both these functions for values 101 and 111. Indicate what values will these functions return?

#include <stdio.h>

#define \_BV( x ) ( 1 << x )

unsigned char countones ( unsigned char ) ;

unsigned char countzeros ( unsigned char ) ;

void main()

{

int num, zeros, ones ;

zeros = countzeros ( 101 ) ;

ones = countones ( 101 ) ;

printf( "0's = %d 1s = %d\n", zeros, ones ) ;

zeros = countzeros ( 111 ) ;

ones = countones ( 111 ) ;

printf( "0's = %d 1s = %d\n", zeros, ones ) ;

}

unsigned char countones ( unsigned char n )

{

int ones = 0 ;

int i ;

unsigned char j, k, andmask ;

for ( i = 7 ; i >= 0 ; i-- )

{

j = i ;

andmask = \_BV( j ) ;

k = n & andmask ;

if ( k == andmask )

ones++ ;

}

return ( ones ) ;

}

unsigned char countzeros ( unsigned char n )

{

int zeros = 0 ;

int i ;

unsigned char j, k, andmask ;

for ( i = 7 ; i >= 0 ; i-- )

{

j = i ;

andmask = \_BV( j ) ;

k = n & andmask ;

if ( k != andmask )

zeros++ ;

}

return ( zeros ) ;

}

Q200. Write a program to find maximum out of three given numbers in a single statement. What are the pros and cons of using this statement?

#include <stdio.h>

void main()

{

int a, b, c, largest ;

printf ( "Enter three numbers: " ) ;

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

largest = a > b ? ( a > c ? a : c ) : ( b > c ? b : c ) ;

printf ( "%d is the largest number.", largest ) ;

}

Pro - The program becomes compact

Con - Nested conditional operators are difficult to understand and debug.