**Programming in C Quiz Set 3:**

1. **void main()**

**{**

**printf (“DeshDeepak”+5);**

**printf (“123456”+2);**

**}**

1. **void main()**

**{char a[]={37, 111, 32, 37, 120, 0};**

**printf (a, a[0], a[1]);**

**}**

1. **void main()**

**{**

**char dest[]= “Visual basic”;**

**char source[]= “C++”;**

**puts(strcpy(&dest[7], source));**

**}**

1. **void main()**

**{**

**char a[5][20]={“Maa ganga”, “jamuna”, “Kaveri”, “Godawari”, “Gomti”};**

**int i; char \*t;**

**t=a[0];**

**while(\*t++!=32);**

**{**

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

**{**

**puts(a[0]);**

**strcpy(t, a[i+1]);**

**}**

**}**

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

**{**

**int i;**

**printf (“argc count=%d\n”, argc);**

**printf (“argv content\n”);**

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

**{**

**printf(“%s\n”, argv[i]);**

**}**

**}**

Consider Input in Command Prompt: C:\>desh Digital India

Find output.

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

**{**

**int data;**

**data=argv[1]+argv[2]+argv[3];**

**printf(“%d”, data);**

**}**

Consider input in Command Prompt: C:\>desh123 1 2 3

Find output.

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

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

**{**

**int data;**

**data=atoi(argv[1])+atoi(argv[2])+atoi(argv[3]);**

**printf(“%d”, data);**

**}**

Consider input in Command Prompt: C:\>desh123 1 2 3

Find output.

1. **What would be the size of the following arrays?**

char s[ ]= “Deepak”;

char s1[ ]= {‘D’, ‘e’, ‘e’, ‘p’, ‘a’, ‘k’};

1. **Is the declaration** char s[ ]= “Deepak” **same as**

char \*str= “Deepak” **? Explain.**

1. **char \*display ()**

**{ char s[ ]= “Desh”;**

**puts(s);**

**return s;**

**}**

**main ()**

**{ puts(display());**

**}**

1. **char \*display ()**

**{ static char s[ ]= “Desh”;**

**puts(s);**

**return s;**

**}**

**main ()**

**{ puts(display());**

**}**

1. **What do you mean by dangling pointers?**
2. **char \*display ()**

**{ char \*s= “Desh”;**

**puts(s);**

**return s;**

**}**

**main ()**

**{ puts(display());**

**}**

1. **#define sqr(x) x\*x**

**void main ()**

**{ float r;**

**r=1.0/sqr(2);**

**printf (“%f”, r);**

**}**

1. **#define sqr(x) (x\*x)**

**void main ()**

**{ float r;**

**r=1.0/sqr(2);**

**printf (“%f”, r);**

**}**

1. **#define sqr(x) ((x)\*(x))**

**void main ()**

**{ int num=3, r;**

**r=sqr(num+1);**

**printf (“%d”, r);**

**}**

1. **#define sqr(x) (x\*x)**

**void main ()**

**{**

**int num=3, r;**

**r=sqr(num+1);**

**printf (“%d”, r);**

**}**

1. **#define sqr(x) (x)\*(x)**

**void main ()**

**{**

**int num=3, r;**

**r=sqr(++num);**

**printf (“%d”, r);**

**}**

1. **main ( )**

**{ char \*s= “Desh”;**

**printf (s “Deepak”);**

**}**

1. **#define sqr (x) x\*x**

**main ()**

**{**

**printf(“value=%d”, sqr(2));**

**}**

1. **#define + -**

**#define \* /**

**main ()**

**{ int a;**

**a= 2+3\*5;**

**printf(“%d”, a);**

**}**

1. **#define IND(x) #x**

**#includeIND(stdio.h)**

**main ()**

**{ printf(“Hi”);**

**}**

1. **#define CONST 100**

**main ()**

**{**

**#define CONST 10**

**printf ( “%d”, CONST);**

**}**

1. **enum CAR {alto, omni, honda=3, wagonR, swift=1, audi};**

**main ()**

**{**

**printf (“%d %d%d %d%d %d”, alto, omni, honda, wagonR, swift, audi);**

**}**

1. **enum COLOR {red, green=red, blue=green};**

**main ()**

**{ printf (“%d %d %d”, red, green, blue);**

**}**

1. **#define struct union**

**struct type**

**{ char a;**

**int b;**

**float c;**

**};**

**main ()**

**{ printf (“%d”, sizeof(struct type));**

**}**

1. **typedef struct union**

**struct type**

**{ char a;**

**int b;**

**float c;**

**};**

**main ()**

**{ printf (“%d”, sizeof(struct type));**

**}**

1. **What do you mean by memory leaks and Buffer Overflow?**
2. **What is a generic pointer?**
3. **Compare Static vs Dynamic memory allocation.**

**Coding Problem:**

1. Median in a sorted 1-D integer array
2. Equilibrium Point in 1-D integer array(**sum of left side elements should be equal to sum of right hand side elements**)
3. All Permutation of a given input string
4. Finding first three greatest numbers in a given input 1-D array
5. A sorted array of integers and a number ***m*** is given; find the closest number to ***m***.