Department of Computer Science & Engineering
Indian Institute of Technology Kharagpur

Practice Sheet #07

Topic: Pointer in C

Date: 23-02-2017

1. Assume the following C variable declaration

   int *A [10], B[10][10];

   Of the following expressions I A[2] II A[2][3] III B[1] IV B[2][3] which will not give compile-time errors if used as left hand sides of assignment statements in a C program?

   A. I, II, and IV only

   B. II, III, and IV only

   C. II and IV only

   D. IV only

2. The following program fragment is written in a programming language that allows variables and does not allow nested declarations of functions.

```
global int i = 100, j = 5;
void P(x)
{
   int i = 10;
   print(x + 10);
   i = 200;
   j = 20;
   print(x);
}
main()
{
   P(i + j);
}
```

   If the programming language uses static scoping and call by need parameter passing mechanism, the values printed by the above program are

   A. 115, 220

   B. 25, 220

   C. 25, 15

D. 115, 105

3. The following program fragment is written in a programming language that allows variables and does not allow nested declarations of functions.

```
global int i = 100, j = 5;
void P(x)
{
   int i = 10;
   print(x + 10);
   i = 200;
   j = 20;
   print(x);
}
main()
{
   P(i + j);
}
```

If the programming language uses dynamic scoping and call by name parameter passing mechanism, the values printed by the above program are :

A. 115, 220

B. 25, 220

C. 25, 15

D. 115, 105

4. Consider the C program shown below.

```
#include <stdio.h>
#define print(x) printf("%d ", x)
int x;
void Q(int z)
{
   z += x;
   print(z);
}
void P(int *y)
{
   int x = *y + 2;
   Q(x);
   *y = x - 1;
   print(x);
```

```
}
main(void)
{
   x = 5;
   P(&x);
   print(x);
}
```

The output of this program is

A. 12 7 6

B. 22 12 11

C. 14 6 6

D. 7 6 6

5. What does the following C-statement declare?

```
int ( * f) (int * ) ;
```

A. A function that takes an integer pointer as argument and returns an integer.

B. A function that takes an integer as argument and returns an integer pointer.

C. A pointer to a function that takes an integer pointer as argument and returns an integer.

D. A function that takes an integer pointer as argument and returns a function pointer

6. Consider this C code to swap two integers and these five statements after it:

```
void swap(int *px, int *py)
{
  *px = *px - *py;
  *py = *px + *py;
  *px = *py - *px;
}
```

S1: will generate a compilation error S2: may generate a segmentation fault at runtime depending on the arguments passed S3: correctly implements the swap procedure for all input pointers referring to integers stored in memory locations accessible to the process S4: implements the swap procedure correctly for some but not all valid input pointers S5: may add or subtract integers and pointers.

A. S1

B. S2 and S3

C. S2 and S4

D. S2 and S5

7. What is printed by the following C program?

```
include <stdio.h>
int f(int x, int *py, int **ppz)
{
 int y, z;
 **ppz += 1;
  z  = **ppz;
 *py += 2;
  y = *py;
  x += 3;
  return x + y + z;
}

void main()
{
  int c, *b, **a;
  c = 4;
  b = &c;
  a = &b;
  printf( "%d", f(c,b,a));
  getchar();
}
```

A. 18

B. 19

C. 21

D. 22

8. Output of following program?

```
#include <stdio.h>
int fun(int n, int *f_p)
{
   int t, f;
   if (n <= 1)
   {
      *f_p = 1;
      return 1;
   }
   t = fun(n- 1,f_p);
```

```
    f = t+ * f_p;
    *f_p = t;
    return f;
}

int main()
{
    int x = 15;
    printf (" %d \n", fun(5, &x));
    return 0;
}
```

A. 6

B. 8

C. 14

D. 15

9. What does the following program print?

```
#include
void f(int *p, int *q)
{
  p = q;
 *p = 2;
}
int i = 0, j = 1;
int main()
{
  f(&i, &j);
  printf("%d %d \n", i, j);
  getchar();
  return 0;
}
```

A. 2 2

B. 2 1

C. 0 1

D. 0 2

10. #include<stdio.h>

```
int f(int *a, int n)
{
  if(n <= 0) return 0;
```

```
  else if(*a % 2 == 0) return *a + f(a+1, n-1);
  else return *a - f(a+1, n-1);
}

int main()
{
  int a[] = {12, 7, 13, 4, 11, 6};
  printf("%d", f(a, 6));
  getchar();
  return 0;
}
```

A. -9

B. 5

C. 15

D. 19

11. What is the return value of f(p,p), if the value of p is initialized to 5 before the call? Note that the first parameter is passed by reference, whereas the second parameter is passed by value.

```
int f(int &x, int c) {
   c = c - 1;
   if (c==0) return 1;
   x = x + 1;
   return f(x,c) * x;
}
```

A. 3024

B. 6561

C. 55440

D. 161051

12. The output of the following C program is _____.

```
void f1 (int a, int b)
{
  int c;
  c=a; a=b; b=c;
}
void f2 (int *a, int *b)
```

```
{
 int c;
 c=*a; *a=*b;*b=c;
}
int main()
{
 int a=4, b=5, c=6;
 f1(a, b);
 f2(&b, &c);
 printf (“%d”, c-a-b);
 return 0;
}
```

A. -5

B. -4

C. 5

D. 3

13. What is the output of the following C code? Assume that the address of x is 2000 (in decimal) and an integer requires four bytes of memory.

```
#include <stdio.h>

int main()

{

  unsigned int x[4][3] = {{1, 2, 3}, {4, 5, 6},

                          {7, 8, 9}, {10, 11, 12}};

  printf("%u, %u, %u", x+3, *(x+3), *(x+2)+3);

}
```

A. 2036, 2036, 2036

B. 2012, 4, 2204

C. 2036, 10, 10

D. 2012, 4, 6

14. Consider the following function written in the C programming language. The output of the above function on input “ABCD EFGH” is

```
void foo (char *a)
{
 if (*a && *a != ` `)
```

```
  {
    foo(a+1);
    putchar(*a);
  }
}
```

A. ABCD EFGH

B. ABCD

C. HGFE DCBA

D. DCBA

15. Consider the following C program segment.

```
# include <stdio.h>
int main( )
{
   char s1[7] = "1234", *p;
   p = s1 + 2;
   *p = '\0' ;
   printf ("%s", s1);
}
```

What will be printed by the program?

A. 12

B. 120400

C. 1204

D. 1034

16. Consider the following C program.

```
# include <stdio.h>
int main( )
{
  static int a[] = {10, 20, 30, 40, 50};
  static int *p[] = {a, a+3, a+4, a+1, a+2};
  int **ptr = p;
  ptr++;
  printf("%d%d", prt-p, **ptr};
}
```

The output of the program is _____

A. 140

B. 120

C. 100

D. 40

17. Predict the output of the program

```
int main()
{
char *ptr = "IITKharagpur";
printf("%c\n", *&*&*ptr);

getchar();
return 0;
}
```

A- I
B- II
C- IIT
D- IITK

18 Predict the output of the below code

```
#include<stdio.h>

int *fun()
{

        int x = 5;

        return &x;
}


int main()
{
        int *p = fun();
        fflush(stdin);
        printf("%d", *p);
        return 0;
}
```

A- A garbage Address
B- 5
C- Error at int *p= fun();
D- Error at printf(“%d”,*p);

19. Predict the output of the below code

```
#include<stdio.h>

int *fun()
{
```

```
        static int x = 5;
        return &x;
}

int main()
{
        int *p = fun();
        fflush(stdin);
        printf("%d",*p);
}
```

A- A Garbage Address
B- 5
C- Error at int *p=fun();
D- Error at printf(“%d”,*p);

20. Predict the output of the below code

```
#include<stdlib.h>

int main()
{
        int x = 4;
        float y = 5.5;
        void *ptr;
        ptr = &x;
        printf("Integer variable is = %d", *( (int*) ptr) );
        ptr = &y;
        printf("\nFloat variable is= %f", *( (float*) ptr) );
        return 0;
}
```

A. Integer variable is = 4
   Float variable is= 5.500000
B. Integer variable is = 5
   Float variable is= 4.000000
C. Integer variable is = 4
   Float variable is= 5.000000
D. Integer variable is = 5
   Float variable is= 5.500000

21. Predict the output of the below code

```
#include <stdio.h>
int main()
{
        int *ptr = NULL;
        printf("The value of ptr is %u", ptr);
        return 0;
}
```

A. The value of ptr is 0

B. The value of ptr is 1
C. Error at printf statement
D. The value of ptr is a garbage value

22. Predict the output of the below code

```
#include<stdio.h>
int main()
{
        int a = 10;
        void *ptr = &a;
        printf("%d", *ptr);
        return 0;
}
```

A- 10
B- Compile time error
C- A garbage value
D- Print 10 in binary format

23. Predict the output of the below code

```
#include<stdio.h>
int main()
{
        int a = 10;
        void *ptr = &a;
        printf("%d", *(int *)ptr);
        return 0;
}
```

A. 10
B. Compile time error
C. A garbage value
D. Print 10 in Binary format

24. Predict the output of the below code

```
 #include<stdio.h>
int main()
{
        int a[2] = {1, 2};
        void *ptr = &a;
        ptr = ptr + sizeof(int);
        printf("%d", *(int *)ptr);
        return 0;
}
```

A. 2
B. 3
C. 1
D. A garbage value

25- Predict the output of the below code

```
#include <stdio.h>
int main()
{
int *i, *j;
int *ii = NULL, *jj = NULL;
if(i == j)
{
printf("This might get printed if both i and j are same by chance.");
}
if(ii == jj)
{
printf("This is always printed coz ii and jj are same.");
}
return 0;
}
```

A- This is always printed coz ii and jj are same.
B- This might get printed if both i and j are same by chance.
C- Error at assignments of ii and jj
D- Error at if(i==j)

26- Predict the output of the below code

```
#include <stdio.h>

void fun1() { printf("Fun1\n"); }
void fun2() { printf("Fun2\n"); }

void wrapper(void (*fun)())
{
        fun();
}

int main()
{
        wrapper(fun1);
        wrapper(fun2);
        return 0;
}
```

A- Fun1
   Fun2
B- Fun2
   Fun1
C- Fun1
   Fun1
D- Fun2
   Fun2

Problems for Programming Practice

(After the successful studies of Lecture 07 (Pointer in C), the students are supposed to solve the following problems in C programming language.)

1. A swap function is an operation to interchange the values in two storage locations. For example, if X = 55 and Y = -100, then after call of Swap(X, Y), the result will be X = -100 and Y = 55 etc.
   a) Define a function ***void Swap(int x, int y)*** to interchange the values in ***x*** and ***y***. You should call the function from main for any two integer values and print the values both from main (before and after the call of *Swap(…)*) and from inside the body of *Swap(…)*.
   b) What modification in *Swap(…)* you should do to realize another swap function say ***void StringSwap(char *s1, char* s2)*** to interchange two strings ***s1*** and ***s2***. Repeat the same execution of your program to print the output as asked in A6/Q 1(a).

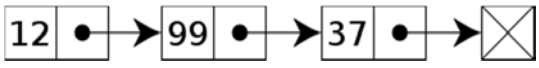
2. A dynamic array is an array whose size is only known during the runtime of a program, which uses the array.
   a) Define an array say **varArray[…]** to store any number of integer values. Initialize the array so allocated and print the array then.
   b) Write a function ***void countPrime(int*, int)***, which receives an integer array and its size, and returns the number of prime numbers in the array.

3. A word is defined as a string of alphabets that does not contain a blank or any special characters (such as , ; . \n etc.). A sentence is considered as a sequence of words separated by blanks and is terminated by '.' , '?' or '\n' character. Write a function ***int WordCount(char* s, char *w)*** that counts the number of occurrences of a word ***w*** in a sentence ***s***. Use pointers to store sentence and word of any size.

4. You are to read any two integer values ***m*** and ***n*** from the keyboard. Then allocate the memory to store a two-dimensional matrix of say ***DynamicMatrix[m][n]***. Initialize the matrix so defined and print the matrix in the matrix form.

5. Read a list of names (First name + Last name and with maximum 15 characters in each) from the keyboard and sort them in ***alphabetical order***.
   You should use the solution in A6/Q 4. to store the list of any number of names, and the size of the list is known while the program runs and A6/Q 1(b) for swapping two strings during sorting procedure.

6. A singly linked list whose nodes contain two fields: an integer value and a link to the next node. An example of such a list is shown in the figure below.



   You have to store a list of numbers using a single linked list structure.

a) Define a structure definition say ***struct Node*** for a node in single linked list.
b) ***struct *Node CreateList(int n)***: To read ***n*** numbers from the keyboard and store them in a single linked list.
c) ***void PrintList(struct *Node myList)*** : Print the elements in the list ***myList***.

7. The user is asked to input two n-dimensional vectors, where each vector is stored by allocating space with the help of pointers. Define a structure to store such a vector.

   Write the C-functions to find the following.
   a) ***struct Vector VectorSum(struct *v1, *v2)***: The sum of the two vectors of the same dimension.
   b) ***float Magnitude(struct *v)***: The magnitude of a vector ***v***.
   c) ***int OrthogonalTest(struct *v1, *v2)***: Whether two vectors ***v1*** and ***v2*** are orthogonal.

   The result computed by each function should return the result to the main program from where the result should be printed.

8. A set can be represented by a dynamic array of elements, where no repetition is permitted. Write C functions to perform the following operations on sets of integer valued elements.
   a) ***int* BuildSet(int n)***: Read ***n*** number of elements and store them in a set.
   b) ***int SearchSet(int* A, int x)***: Search the set ***A*** to find if an element ***x*** is in it.
   c) Given two sets, compute the following.
      i. ***int* Union (int *A, int *B)***: To return the union of two sets ***A*** and ***B***.
      ii. ***int* Intersection (int *A, int *B)***: To return the intersection of two sets ***A*** and ***B***.
      iii. ***int* Difference (int *A, int *B)***: To return the difference of two sets ***A*** and ***B***.
      iv. ***struct OrderPair* CProduct (int *A, int *B)***: To return the Cartesian product of two sets ***A*** and ***B***. [Hint: Define a structure to store an order pair.]

   The main program should call these functions. All results should be stored in their resultant sets and then display the results. [Hint: Define a function ***void Print(int *A)*** to print a set A.]