Week Lab 1

**1. Give memory snapshots after each of these sets of statements is executed:**

**a.**

int a=1, b=2, \*ptr;

…

ptr = &b;

*ANS>*

a b

2

1

1000 2000

2000

Ptr

3000

a = 1

b = 2

O/P

**b.**

int a=1, b=2, \*ptr=&b;

...

a = \*ptr;

ptr = &b;

*ANS>*

a b

**2**

**~~1~~  2**

1000 2000

2000

Ptr

O/P \*ptr = \*(2000)

a = 2

b = 2

**c.**

int a=1, b=2, c=5, \*ptr=&c;

...

b = \*ptr;

\*ptr = a;

ptr = &b;

*ANS>*

a b c

**~~3~~ 1**

**~~2~~ 5**

1

1000 2000 3000

3000

Ptr

4000

B = 5

b = \*btr

C = 1

\*ptr = a \* (300) = 1

a = 1

b = 5

c = 1

O/P

**d.**

int a=1, b=2, c=5, \*ptr;

...

ptr = &c;

c = b;

a = \*ptr;

*ANS>*

a b c

**~~3~~ 2**

**2**

**~~1~~ 2**

1000 2000 3000

3000

\*ptr

4000

a = 2

b = 2

c = 2

O/P

**e.**

You may be asked to demonstrate/explain your work to the tutor, if you are absent/unavailable or fail to demonstrate properly, zero marks will be awarded.

double x=15.6, y=10.2, \*ptr\_1=&y, \*ptr\_2=&x;

...

\*ptr\_1 = \*ptr\_2 + x;

*Ans>*

x y

10.2

15.6

1000 2000

2000

\*ptr\_1

3000

X = 15.600000

Y = 10.199998

O/P

**f.**

int w=10, x=2, \*ptr\_2=&x;

...

\*ptr\_2 -= w;

*Ans>*

w x

**~~2~~ 10**

10

1000 2000

2000

\*prt\_2

3000

w = 10

x = 10

O/P

**2. Assume that an array g is defined with the following statement:**

int g[]={2,4,5,8,10,32,78};

int \*ptr1=&g[0], \*ptr2=&g[3];

Give a diagram of the memory allocation, including the array values. Also indicate the offset values from the initial value in the array. Using this information, give the value of the following references:

The memory diagram of the array is:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **2** | **4** | **5** | **8** | **10** | **32** | **78** |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 |

a. \*g = 2 (Since g gives the address of the first element)

b. \*(g+1) = 4 (points of second elements)

c. \*g+1 = (2) + 1 = 3

d. \*(g+5) = 32

e. \*ptr1 = 2

f. \*ptr2 = \*(g+3) = 8

g. \*(ptr1+1) = 4

h. \*(ptr2+2) = \*(g+3+2) = 32

**3. Assume that an integer array x is defined by the following statements:**

int x[2][4]={{1,8,7,6},{2,4,-1,0}}, \*xptr=&x[0][0];

Draw a memory allocation diagram, and give the value indicated by each of the following references:

The memory diagram of the array

0 1

0 1 2 3 4 5 6 7

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 8 | 7 | 6 | 2 | 4 | -1 | 0 |

x

\*ptr

2000

100

100

1000

a. \*xptr = \*(100) = 1

b. \*(xptr+2) = \*(xptr + 2)

= \*(100 + 2 (2))

= \*(104)

= 7

c. \*xptr + 2 =\*(100 + 2)

= 1 + 2

= 3

d. \*(xptr+1) + \*(xptr+3) =\*(100 + 2(1) + \*(\*100+3(2))

= (\*102) + (\*106)

= 8+6

= 14

**4. A text file contains grade details of five students. This file formatted like name followed by marks for four subjects. A content of a sample text file is given below:**

Peter

55

66

44

67

Lilly

100

90

43

89

John

34

56

78

65

Mary

45

56

78

90

Alex

30

45

65

54

**Write a c program that reads the contents of the file and display it on the screen. Use the following structure for this program.**

struct student

{

char name[10];

int marks[4];

};

**Code:**

#include <stdio.h>

int main()

{

FILE \*ptoFile = fopen("student.txt", "r");

char name[20];

int marks[5];

while (fgets(name, marks, ptoFile) ){

printf(name, marks);

}

printf("\n End of program\n");

fclose(ptoFile);

return 0;

}

**student.txt**

Peter

55

66

44

67

Lilly

100

90

43

89

John

34

56

78

65

Mary

45

56

78

90

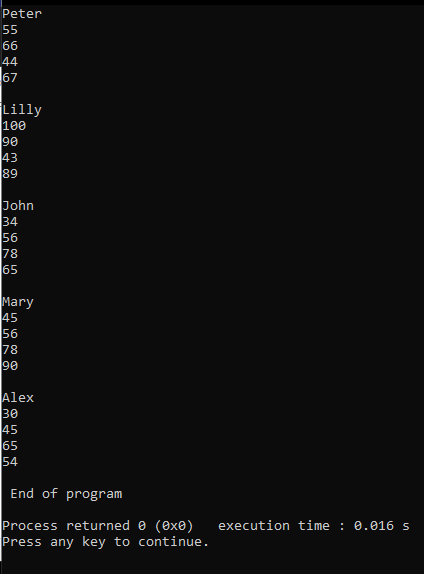
Alex

30

45

65

54

**Output:**