Problem Solving Through Programming in C

Tutorial Session 11

Prof. Anupam Basu
Dept. of Computer Science & Engg.
IIT Kharagpur

Siddhant Mohapatra
PMRF Scholar
IIT Madras

ptr → 0 1 2 3 4

0 2 2 3 4 ← ptr

memory allocation occurs at runtime

int a = 50; → allocated at compile time

malloc } define & initialize memory
calloc }

realloc } reallocate memory.

Dynamic memory allocation

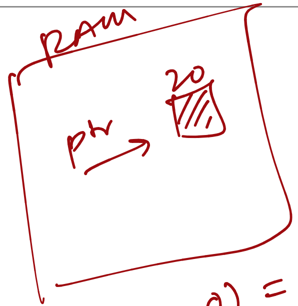
Q: What will be output of the program?

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
   int i;
   int *ptr = (int *) malloc(5 * sizeof(int));

   for (i=0; i<5; i++)
      *(ptr + i) = i;
   printf("%d ", *ptr++);
   printf("%d ", (*ptr)++);
   printf("%d ", *ptr);
   printf("%d ", *++ptr);
   printf("%d ", ++*ptr);
  return 0;
}
```

pointer of integer type

=4

20 bytes



*(ptr+0) = 0 i=0
*(ptr+1) = 1 i=1
*(ptr+2) = 2 i=2
*(ptr+3) = 3 i=3
*(ptr+4) = 4 i=4

++, -- order of procedure

*

++ptr, *ptr

++ptr, *ptr

a) Error
b) 0 1 2 3 4
c) 1 2 3 4 5
[✓ d) 0 1 2 2 3]

Q. The program will allocate …….bytes to ptr. Assume sizeof(int)=4.

```
#include<stdio.h>
#include<stdlib.h>
 int main()
 {
    int *ptr;
    ptr = (int*)malloc(sizeof(int)*4);
    ptr = realloc(ptr,sizeof(int)*2);
    return 0;
 }
```

ptr = (int *)calloc(4, sizeof(int));

16 bytes

8 bytes

↓ 8 bytes of memory is freed.

a) 16
b) 4
[✓ c) 8]
d) None

(datatype*) malloc (no. of bytes to be allocated)

(datatype*)calloc(no. of memory blocks, size of each block)
↳ initializes the value to 0.

Structures

Q. What is the output?

```
#include<stdio.h>
int main()
{
struct xyz{ int a;};
struct xyz obj1={11};
struct xyz obj2 = obj1;
printf("%d", obj2.a);
obj2.a = 101;
printf("%d", obj1.a);
printf("%d", obj2.a);
return 0;
}
```

member of structure xyz.
xyz.a
obj1.a = 11
structure definition.
defining & initializing a structure variable of type struct xyz.
obj2.a = obj1.a = 11
11
11
101

a) 1111011
[✓ b) 1111101]
c) 1110111
d) 1110011

Q. What is the output?

```
#include<stdio.h>
int main()
{
struct xyz{ int a;};
struct xyz obj1={1};
struct xyz obj2 = obj1;
printf("%d", obj2.a);
obj2.a = 100;
printf("%d", obj1.a);
printf("%d", obj2.a);
return 0;
}
```

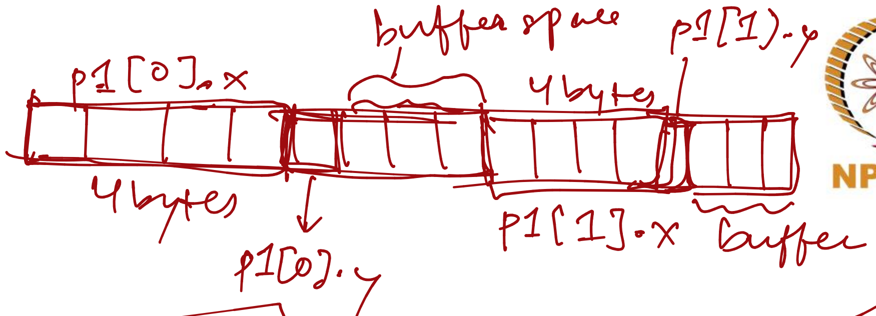
struct xyz with member 'a'.
obj1.a = 1
obj2.a = obj1.a

[✓ a) 11100]
b) 11
c) 11001
d) 11000

Lets say:
sizeof (int) = 4 bytes
sizeof (char) = 1 byte.

Miscellaneous

p1[0].x = 1 ;
p1[0].y = (char)90;
p1[1].x = 62
p1[1].y = (char)33;
p1[2].x = 3
p1[2].y = (char) 34



union → it takes the size of the largest element

Q. What is the output of the following C program?

```
#include <stdio.h>
struct p
{
    int x;
    char y;
};

int main()
{
    struct p p1[] = {1, 90, 62, 33, 3, 34};
    struct p *ptr1 = p1;
    int x = (sizeof(p1) / 3);
    if (x == sizeof(int) + sizeof(char))
        printf("True");
    else
        printf("False");
    return 0;
}
```

2 members, 1 integer and 1 character.

array of structures

p1[0].x p1[0].y p1[1].x

ptr1 which points to the first member of the first structure.

→ 4 → 1

a) True
[✓ b) False]
c) No output
d) Compilation error

size of a structure ≥ sum of size of its members.

8 == 5

Q. Find the output of the following program

```
#include<stdio.h>
int main()
{
    char A[ ] = {'a','b','c','d','e','f','g','h'};
    char *p = A;
    ++p;
    while(*p != 'e')
    printf("%c", *p++);
    return 0;
}
```

p

++p; p

1st iteration
'b' != 'e' true.
p++, *p p → 'c'

2nd iter
true 'c' != 'e'
p++, *p p → 'd'

3rd iter
true 'd' != 'e'
p++, *p, p → 'e'

4th iter
false 'e' != 'e'

a) abcd
[✓ b) bcd]
c) cd
d) abcdfgh

Miscellaneous

"r" → reading mode.
"w" → writing mode.
"a" → appending mode.

fclose → to close the file.

Q. What does fp point to in the program?

```
#include<stdio.h>
int main()
{
   FILE *fp;
   fp=fopen("hello", "r");
   return 0;
}
```

open and access a file

mode

name of the file.

defines a structure with a char pointer as member.

first character of file.

a) The first character in the file
[✓ b)] A structure which contains a char pointer which points to the first character of a file
c) The name of the file
d) The last character in the file

Q. Choose a correct statement about C structure?

```
int main()
{
struct hello{};
return 0;
}
```

a) It is wrong to define an empty structure
[✓ b)] Member variables can be added to a structure even after its first definition
c) There is no use of defining an empty structure
d) None of the above