

# PROBLEM SOLVING WITH C – QUIZ 3

UE17CS151

```
#include<stdio.h>
int main()
{
    int A[][2]={23,22},{12,13},{16,178}};
    printf("%d",1[A][2]);
    return 0;
}
```

What gets printed??

- A) 16
- B) CompileTime Error
- C) Runtime Error
- D) 12

Answer:

- A) 16

```
#include<stdio.h>
int main()
{
    int A[][2]={23,22},{12,13},{16,178}};
    printf("%d",1[2][A]);
    return 0;
}
```

What gets printed??

- A) 16
- B) CompileTime Error
- C) Runtime Error
- D) 12

Answer:

- B) CompileTime Error

```
#include<stdio.h>
int main(){
    int A[][2]={23,22,12,13,16,178};
    printf("%d %d",A[2][1], A[1][2]);
    return 0;
}
```

What gets printed??

- A) 178 16
- B) 178 178
- C) 16 178
- D) 12 23

Answer:

- A) 178 16

# What will be output of the following program?

```
#include<stdio.h>
#pragma pack(1)
struct testing
{
    int i;           // size of int is 4 bytes
    char c;          // size of char is 1 byte
    float f; // size of float is 4 bytes
};
int main()
{
    printf("%d",sizeof(t));
}
```

Answer:  
9

# What gets printed?

```
#pragma pack(1);  
struct test  
{ float f; char r; };// size of float is 4 bytes, sizeof char is 1 byte  
int main()  
{  
    printf("%lu",sizeof(struct test *));  
}
```

- A) 4
- B) depending on the machine architecture
- C) 8
- D) 5

**Answer:** B) depending on the machine architecture

# What gets printed?

```
struct test
{ float f; char r; };
int main(){
struct test t2; struct test t1;
t2.f=23.7;
t2.r='p';
t1=t2;
printf("%d",t2==t1);
}
```

- A) Compilation Error
- B) 1
- C) 0
- D) True

Answer:

A) Compilation Error  
== cannot be used on Structure variables

If t is pointing to 2008, What gets printed?

```
#pragma pack(1)
```

```
struct test
```

```
{    float f;    char r;    }; // size of float = 4 bytes
```

```
#include<stdio.h> // size of char = 1 byte
```

```
int main()
```

```
{    struct test *t;
```

```
    t++;
```

```
    printf("%p",t);
```

```
    return 0;
```

```
}
```

Answer:

A) 2013



If td is a pointer to structure and structure is declared as below, how to access the members of the array using td?

```
struct test_data
{
    int d;
    double data;
};
```

Answer: td->d and td->data

OR

(\*td).d and (\*td).data

# What is the output of code?

```
#include<stdio.h>

struct test
{
    int i;    double d;    }t1;

int main()
{
    printf("%d %lf",t1.i,t1.d);
    t1.i=23;t1.d=23.7;    }
```

- A) Junk values
- B) 0 0.000000
- C) 23 23.700000
- D) Compile time Error

Answer:

B) 0 0.000000

# What is the return type of malloc()?

Answer :

`void *`

`malloc()` returns generic pointer

is the unorganized segment of process memory used for dynamic memory allocation

- A) Stack
- B) Heap
- C) Text Segment
- D) None of these

Answer: B) Heap

```
int i=50;  
const int * const p;  
p=&i;  
printf("%f\n",*p);
```

- A) 50.000000
- B) Compilation Error
- C) 50
- D) Runtime Error

Answer: B) Compilation Error

```
int i=50;  
const int * const p=&i;  
printf("%d\n",*p);
```

- A) 50.000000
- B) Compilation Error
- C) 50
- D) Runtime Error

Answer: C) 50

```

struct test
{
    int t;
    float r;
};

void f1(struct test *t)
{
    int i = 0;
    while(i<2)
    {
        t++;          i++;
    }
    printf("%d %f\n",t->t,t->r);
}

struct test t[10]={{1,2.5},{9,1.3}};
f1(t);

```

- A) 1 2.500000
- B) Undefined value
- C) 0 0.000000
- D) Compiletime Error

Answer: C) 0 0.000000

Partial initialization of the array results in default values for explicitly uninitialized memory locations

```
typedef struct test
{
    int a;
    struct test *t;
}t1;
int main()
{
    t1.a = 100;
    t1.t = malloc(sizeof(struct test));
    t1.t->a = t1.a;
    printf("%d", t1.t->a);
}
```

- A) Prints undefined value
- B) Prints 100
- C) Results in Compilation Error
- D) Results in segmentation Fault

Answer: C) Results in Compilation Error  
t1 is an alias for the type. Not a variable



```
struct test
{
    int a;
    struct test *t;
}t1;
int main()
{
    t1.a = 100;
    t1.t = malloc(sizeof(struct test));
    t1.t->a = t1.a;
    printf("%d", t1.t->a);
}
```

- A) Prints undefined value
- B) Prints 100
- C) Results in Compilation Error
- D) Results in segmentation Fault

Answer: B) Prints 100

t1 is a variable of type struct test

```
struct site {  
    char name[] = "pes";  
    int no_of_pages = 200;  
};  
int main()  
{  
    struct site *ptr;  
    printf("%d ", ptr->no_of_pages);  
    printf("%s", ptr->name); }  
What Is the output of above code?
```

- A) 200 pes
- B) Prints 200 and then segmentation Fault
- C) Results in Compilation Error
- D) Results in segmentation Fault

Answer: C) Results in Compilation Error

Data members cannot be initialized inside the structure

```
struct test { int i; float j; };  
struct test *s[10];  
choose the correct statement .
```

- A) s is an array of pointer to structure
- B) s is a pointer to structure
- C) s is a structure
- D) s is array of structure

Answer: A) s is an array of pointer to structure

```
struct test
{
    int a;
    int b;
}t1;
```

// t1 is a global variable

Which statements are true?

- A) t1.a is 0
- B) t1 is the alias for struct test
- C) t1.a is an undefined value
- D) Compiletime Error
- E) None of these

Answer: A) t1.a is 0

```
char *p[2]={"xyz","lmn"};  
printf("%s",*p);
```

What is the output?

- A) x
- B) xyz
- C) Compile-time Error
- D) Runtime Error

Answer: B) xyz

```
char *p[2]={"xyz","lmn"};  
printf("%s",*p+1);
```

What is the output?

- A) xy
- B) xyz
- C) yz
- D) Compiletime Error

Answer: C) yz

```
char *p[2]={"xyzabc","lmn"};  
printf("%s",*(p+1));
```

What is the output?

- A) l
- B) xyz
- C) Compile-time Error
- D) lmn

Answer: D) lmn

```
char *p[2]={"xyzabc","lmn"};  
printf("%s",*p+1);
```

What is the output?

- A) Any junk value
- B) xyzab
- C) yzabc
- D) lmn

Answer: C) yzabc



```
char *p[4]={"xyz","lmn","pqr","abc"};  
char **pp[4] = {p+2,p+3,p,p+1};  
printf("%s",**(pp+2));
```

What is the output?

- A) Undefined value
- B) xyz
- C) abc
- D) pqr

Answer: B) xyz

```
char *p[4]={"xyz","lmn","pqr","abc"};  
char **pp[] = {p+2,p+3,p,p+1,p,p+2,p+1,p};  
printf("%s",**(pp+7));
```

What is the output?

- A) Compile-time Error
- B) xyz
- C) Undefined value
- D) Seg Fault

Answer: B) xyz

Write the code snippet to assign the values to the members of the structure through t.

```
struct Test
{
    int i; int *p; float j;
};
int main()
{
    struct Test *t;
    // write code here
}
```

Answer: t->i=23; t->p=&(t->i);t->j=92.5;

Write the code snippet to print the value of i using p and p1 both.

```
int i=20;  
int *p = &i;  
int *p1=p;
```

Answer: `printf("%d %d",*p,*p1);`

Is there any static allocation of memory in the below C statement?

```
int *p1=(int *) malloc(3*sizeof(int));
```

Answer: Yes. For p , static allocation of memory is done

```
typedef struct NODE
{
    int i;
    int j;
    struct NODE next;
}NODE;
```

- A) Runtime Error
- B) Machine Dependent
- C) Compile time Error
- D) Only Warning

Answer: C) Compile time Error

Reason: field 'next' has incomplete type

```
struct Test{    int data;        struct Test *next;};  
typedef struct Test TEST;  
int main()  
{  
    TEST t;  
    t.next = (TEST*)malloc(sizeof(TEST));  
    t.next->next = (TEST*)malloc(sizeof(TEST));  
    t.next->data = 40;  
    printf("%d",t.next->data);  
}
```

What is the output of above code?

- A) Runtime Error
- B) 40
- C) Compile time Error
- D) Undefined Value

Answer: B) 40

```
struct Test{    int data;        struct Test *next;};  
typedef struct Test TEST;  
int main()  
{  
    TEST t;  
    t.next = (TEST*)malloc(sizeof(TEST));  
    t.next->next = (TEST*)malloc(sizeof(TEST));  
    t.next->data = 40;  
    free(t.next); printf("%d",t.next->data);  
}
```

Choose the correct statement about the above code.

- A) Above code will result in seg fault
- B) Above code could result in seg fault
- C) Compile time Error
- D) None of the options are true.

Answer: B) Above code could result in seg fault



Which of the following is True?

- A) Dangling pointers are dangerous
- B) Creation of garbage is not a good programmer's skill.
- C) Both B and D
- D) Dereferencing dangling pointer is dangerous
- E) Both A and B

Answer: C) Both B and D

Which of the following operators can be applied on structure variables?

- A) Equality comparison ( == )
- B) Assignment ( = )
- C) Both of the above
- D) None of the above

Answer: B) Assignment ( = )

A new type Test is created with str as an array of characters, data member of the type.  
str is big enough to hold maximum of 8 characters.

```
struct Test st1, st2;  
strcpy(st1.str, "pes");  
st2 = st1; st1.str[0] = 'S';  
printf("%s %s",st1.str, st2.str);
```

- A) Assignment operator cannot be used with structure variables
- B) Sespes
- C) pespes
- D) SesSes

Answer: B) Sespes

Choose the correct statement about s.

```
struct test{ int i; float j;};struct test *s[10] ;
```

- A) A structure of 2 fields, each field being a pointer to an array of 10 elements.
- B) A structure of 3 fields: an integer, a float, and an array of 10 elements
- C) An array, each element of which is a pointer to a structure of type test
- D) An array, each element of which is a structure of type test

Answer: C) An array, each element of which is a pointer to a structure of type test

```
int a[5] = {12,89,45};  
int *p[5]; int i;  
for(i = 0;i<5;i++)  
    p[i] = &a[i];  
printf("%d ",*(p[i-1]));  
What gets printed?
```

- A) Compilation Error
- B) 89
- C) 45
- D) 0

Answer: D) 0

```
typedef struct NODE
{
    int i;   int j;   struct NODE *next;
}NODE;
#include<stdio.h>
int main()
{
    NODE temp1;
    struct NODE temp2;
    printf("%d %d",temp1.i,temp2.j);
    return 0;
}
```

Answer:

- A) Runtime Error
- B) Compilation Error
- C) Undefined values are printed
- D) 0 0

```
struct Student
{
    char name[20];
    int srn;
    struct DOB {
        int year; int month; int day;
    } *d;
};
```

Accessing the year through s1, where s1 is the instance variable of struct Student

- A) s1.d.year
- B) s1.d->year
- C) Both A and B
- D) Cant access year through s1

Answer: B) s1.d->year

```
struct Student
{
    char name[20];      int srn;
    struct DOB
    {
        int year;      int month;      int day;
    };
}s1;
```

If s1 is a structure variable outside main, What gets printed when we print month using s1.month?

- A) 0      B) cannot access month using s1. So Compile time Error  
B) Undefined value      D) Segmentation Fault

Answer: A) 0



# THANK YOU

For Queries,  
Prof. Sindhu R Pai  
Department of CSE, PESU  
Email: [sindhurpai@pes.edu](mailto:sindhurpai@pes.edu)