

Subject: Programming in C

Arrays and Pointers

DPP-01

[MSQ]

1. Which of the following declarations are INVALID?

- (a) int b[][4];
- (b) int b[];
- (c) int b[2][][2]={1,2,3,4};
- (d) int b[][2][2]={1,2,3,4};

[MCQ]

2. Consider the following two statements:

```
P: int a[3]={1, 2, 3};
printf("%d", *a++);
```

```
Q: int a[3]={1, 2, 3};
int *p=a;
printf("%d", *p++);
```

Which of the following statements is/are CORRECT?

- (a) P only.
- (b) Q only.
- (c) Both P and Q.
- (d) Neither P nor Q.

[MCQ]

3. Consider the following program:

```
#include<stdio.h>
int main(void)
{
    int a[5]={5, 10, 15};
    printf("%d", 1[a]);
    return 0;
}
```

The output is-

- (a) 5
- (b) 10
- (c) Garbage value
- (d) Compilation error

[MCQ]

4. Consider the following program:

```
#include<stdio.h>
int main(void)
{
    int 5[a]={5, 10, 15};
```

```
    printf("%d", 1[a]);
    return 0;
}
```

The output is-

- (a) 5
- (b) 10
- (c) Garbage value
- (d) Compilation error

[MCQ]

5. Consider the following program:

```
#include<stdio.h>
int main(void) {
    int a[5]={5, 10, 15, 20, 25};
    printf("%u", a);
    printf("%u", *(a+3));
    printf("%u", a+2);
    printf("%u", *(a+2)+6);
    printf("%u", *(a+*(a+1)-6));
    return 0;
}
```

Assuming the base address of the array to be 1000 and integer size as two bytes the output is-

- (a) 1000 20 1004 21 25
- (b) 5 20 15 21 25
- (c) 1000 20 1002 21 24
- (d) Compilation error

[MCQ]

6. Consider the following program:

```
#include<stdio.h>
int main(void)
{
    int a[5]={5, 10, 15, 20, 25};
    printf("%u\t", *(1+a));
    printf("%u\t", &a+1);
    return 0;
}
```

Assuming the base address of the array to be 1000 and integer size as four bytes the output is-

- (a) 1004 1020
- (b) 10 1016
- (c) 10 1020
- (d) 1004 1016

Answer Key

- 1. (a, b, c)
- 2. (b)
- 3. (b)
- 4. (d)

- 5. (a)
- 6. (c)



Hints and solutions

1. (a, b, c)

- (a) int b[][][4]: Invalid as elements are not specified.
- (b) int b[]; Invalid as size is not specified.
- (c) int b[2][][]={1,2,3,4}; Invalid. If the elements are specified, only first dimension can be omitted.
- (d) int b[][2][2]={1,2,3,4}; Valid. If the elements are specified, only first dimension can be omitted.

2. (b)

int a[3]={1, 2, 3};

Array name without subscript denotes the base address of the array. So, a++ is not allowed.
Hence, P is incorrect.

Q is correct.

3. (b)

The printf() statement can be interpreted as-

printf("%d", 1[a]) is equivalent to printf("%d", *(1+a));

1000	1002	1004
5	10	15

So, *(1+a) is equivalent to *(1+1000). Here, 1 signifies the increment by 1*2 bytes= 2 bytes.
So, *(1002) is 10.

4. (d)

int 5[a]={5, 10, 15}; // It is an invalid declaration.

So, compilation error will happen.

5. (a)

1000	1002	1004	1006	1008
5	10	15	20	25

printf("%u", a);//1000

printf("%u", *(a+3));//*(1000+2*3) i.e *1006 i.e 20

printf("%u", a+2);//1000+2*2=1004

printf("%u", *(a+2)+6);//*(1000+2*2)+6 i.e *1004+6 i.e 15+6 i.e 21

printf("%u", *(a+*(a+1)-6));

//**(a+*(1000+2*1)-6) = *(a+4) = *(1000+2*4) = *1008 i.e 25

Output: 1000 20 1004 21 25

6. (c)

1000	1004	1008	1012	1016
5	10	15	20	25

printf("%u\t", *(1+a));//*(1*4+1000)=*1004=10 &a+1 signifies the next 1D array. So, size incrementing by 1 means increase by 4*5 bytes.

printf("%u\t", &a+1);//1000+20=1020 is printed.

Output: 10 1020



Any issue with DPP, please report by clicking here:- <https://forms.gle/t2SzQVvQcs638c4r5>

For more questions, kindly visit the library section: Link for web: <https://smart.link/sdfez8ejd80if>



PW Mobile APP: <https://smart.link/7wwosivoicgd4>

Programming in C

Arrays and Pointers

DPP-02

[MCQ]

1. Consider the following program:

```
#include<stdio.h>
int main()
{
    int a[5]={5, 3, 1, 2, 4};
    int *p[5]={a, a+1, a+3, a+2, a+4};
    printf("%u\t%u", p[3][1], *((p+4)-2));
    return 0;
}
```

The output is:

- | | |
|---------|--------------------|
| (a) 5 3 | (b) 4 1 |
| (c) 2 1 | (d) Garbage value. |

[MCQ]

2. Consider the following program:

```
#include <stdio.h>
int main()
{
    int a[]={2, 4, 6};
    int b[]={1, 3, 5};
    int *arr[]={a, b};
    printf("%u\t", *((arr+1)+2));/*line 1*/
    printf("%u\t", **arr+3);/*line 2*/
    printf("%u", ***arr);/*line 3*/
    return 0;
}
```

Which of the following lines does not give ERROR?

- (a) Line 1 only
- (b) Line 1 and Line 2 only
- (c) Line 3 only
- (d) Line 2 and Line 3

[MCQ]

3. Consider the following function:

```
void f(int *p, int n)
{
    static int i;
    i=n-1;
    if(i<0) return;
    printf("%d\t", p[i]+p[n-i-1]);
    f(p,n-1);
}
```

If the array arr with elements [1, 2, 3, 4, 5] is passed as f(arr, 5), the output is-

- | | |
|---------------|---------------|
| (a) 5 4 3 2 1 | (b) 7 6 5 4 3 |
| (c) 6 5 4 3 2 | (d) 2 3 4 5 6 |

[NAT]

4. Consider the following program:

```
#include<stdio.h>
int main()
{
    int a[5]={5, 3, 1, 2, 4};
    int *p[5]={a+3, a+1, a, a+2, a+4};
    int **ptr=p+3;
    printf("%u\t%u\t%u", ptr-p, *ptr-a, **ptr);
    return 0;
}
```

The sum of the output is _____.

5. Consider the following function:

```
void func(int (*ptr) [2])
{
    **ptr+=1;
    ptr++;
    **ptr=3;
}
```

The array arr[2][2] with elements {0, 1, 2, 3} is passed to func(). What are the contents of the array after calling func()?

[MCQ]

6. Consider the following program:

```
#include<stdio.h>
int main()
{
    int a[][3][2]={1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23};
    printf("%u\t", a+1);
    printf("%u\t", *a+1);
    printf("%u\t", **a+1);
    printf("%u\t", ***a+1);
    printf("%u\t", &a+1);
    return 0;
}
```

Assume the base address of a is 100 and integer size is 2 bytes, the output is-

- (a) 124 112 102 2 106
 - (b) 124 102 112 5 106
 - (c) 112 104 102 2 124
 - (d) 112 104 102 5 124

[MCQ]

7. Consider the following program:

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

```
int main()
{
    int a[3][2]={1, 3, 5, 7, 9, 11};
    int *ptr=a;
    ptr+=sizeof(int);
    printf("%d", *ptr);
    return 0;
```

(Assume size of int to be 2 bytes.)

The output is .

8. Consider the following program:

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

```
int main()
```

{

```
int a[][2]={1, 3, 5, 7, 9, 11};  
int *ptr=a[1];  
++*ptr++;  
printf("%d", *ptr);  
return 0;
```

The output is-

Answer Key

Hints and solutions

1. (c)

	1000	1002	1004	1006	1008
a	5	3	1	2	4
	2000	2004	2008	2012	2016

p	1000	1002	1006	1004	1008
---	------	------	------	------	------

$p[3][1]=*(p+3)+1=*(2012+1)=*(1004+1)=1006=2$
 $*(p+4)-2=*(2016-2)=*(1008-2)=1004=1$

Output: 2 1

2. (b)

	1000	1002	1004
a	2	4	6

	2000	2002	2004
b	1	3	5

	3000	3004
arr	1000	2000

$\text{printf}("%u\t", *(arr+1)+2);$ //It prints arr[1][2]
i.e 5

$\text{printf}("%u\t", **arr+3);$ //It prints arr[0][0]+3 i.e 5
 $\text{printf}("%u", ***arr);$ // It gives ERROR as here
dereferencing can happen only at two levels

3. (c)

	1000	1002	1004	1006	1008
a	1	2	3	4	5

f(arr, 5):
p=1000, n=5;
static int i; //i is initialized to 0
i=n-1; //i=4
4<0: FALSE
 $\text{printf}("%d\t", p[i]+p[n-i-1]);$ //p[4]+p[0]=5+1=6 is printed.
f(1000, 4) is called.
f(1000, 4):

i=n-1; //i=3

3<0: FALSE

$\text{printf}("%d\t", p[i]+p[n-i-1]);$ //p[3]+p[0]=4+1=5 is printed.

f(1000, 3) is called.

f(1000, 3):

i=n-1; //i=2

2<0: FALSE

$\text{printf}("%d\t", p[i]+p[n-i-1]);$ //p[2]+p[0]=3+1=4 is printed.

f(1000, 2) is called.

f(1000, 2):

i=n-1; //i=1

1<0: FALSE

$\text{printf}("%d\t", p[i]+p[n-i-1]);$ //p[1]+p[0]=2+1=3 is printed.

f(1000, 1) is called.

f(1000, 1):

i=1-1; //i=0

0<0: FALSE

$\text{printf}("%d\t", p[i]+p[n-i-1]);$ //p[0]+p[0]=1+1=2 is printed.

f(1000, 0) is called. It simply returns.

Output: 6 5 4 3 2

4. (6)

	1000	1002	1004	1006	1008
a	5	3	1	2	4
	2000	2004	2008	2012	2016

p	1006	1002	1000	1004	1008
---	------	------	------	------	------

ptr=p+3

ptr-p=p+3-p=3

$*ptr-a=*(p+3)-1000=2012-1000=(1004-1000)/2=2$

$**ptr=**(p+3)=2012=1004=1$

Output: 3 2 1

Sum: 6

5. (a)

The function increments the 0th element of 0th row by 1 and 0th element 1st row by 3.

Output: 1 1 6 3

6. (c)

a+1 points to the 1st 2D array.

*a+1 points to the 1st 1D array of the 0th 2D array.

**a+1 points to the 1st element of the 0th 1D array of the 0th 2D array.

***a is the the 0th element of the 0th 1D array of the 0th 2D array.

&a+1 is the address of the next 3D array.

Output is:

112 104 102 2 124

7. (5)

ptr stores the address of the 0th 1D array.

ptr+=2 means ptr now points to the 2nd element.

printf("%d", *ptr); //5 is printed.

8. (c)

	1000	1002	1004	1006	1008	1010
a	1	3	5	7	9	11

ptr is a single pointer. So, it increments linearly.

ptr stores the address of (a+1). ptr=1004;

++*ptr++ increments ptr by 1.

ptr=1006

Output: 7



Any issue with DPP, please report by clicking here:- <https://forms.gle/t2SzQVvQcs638c4r5>

For more questions, kindly visit the library section: Link for web: <https://smart.link/sdfez8ejd80if>



PW Mobile APP: <https://smart.link/7wwosivoicgd4>

Programming in C

Arrays and Pointers

DPP-03

[NAT]

1. Consider the following program:

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

The output is _____.

[MCQ]

2. Consider the following program:

```
#include<stdio.h>
int * f(){
    static int a[4]={1, 2, 3, 4};
    return a;
}
int main()
{
    int *p, i;
    p=f();
    for(i=0;i<3;i++){
        printf("%d\t", p[i]+p[i+1]);
    }
    return 0;
}
```

The output is-

- (a) Compilation Error
- (b) Runtime Error
- (c) 3 5 7
- (d) None

[NAT]

3. Consider the following program:

```
#include<stdio.h>
int main()
{
    int p=10, s=20, *q, **r;
    q=&p;
    *q=p++>*q;
    q=&s;
    r=&q;
    **r=--*q***r;
    printf("%d", p+s);
    return 0;
}
```

The output is _____.

[MCQ]

4. Consider the following program:

```
#include<stdio.h>
int * f()
{
    int a[4]={1, 2, 3, 4};
    return a;
}
int main()
{
    int *p, i;
    p=f();
    for(i=0;i<3;i++){
        printf("%d\t", p[i]+p[i+1]);
    }
    return 0;
}
```

The output is-

- (a) compilation Error
- (b) Runtime Error
- (c) 3 5 7
- (d) None

[MCQ]

5. Consider the following statements:

- P: int * p(int *) - p is a function that takes an integer pointer as argument and returns an integer pointer.
 Q: int (*p(int *))[] - p is a function that takes an integer pointer as argument and returns a pointer to an array of integers.

Which of the following is INCORRECT?

- | | |
|------------------|----------------------|
| (a) P only | (b) Q only |
| (c) Both P and Q | (d) Neither P nor Q. |

[MCQ]

6. Consider the following program:

```
#include<stdio.h>
void f(int (*q)[2]){
    printf("%d\t",(*q)[1]);
    q+=2;
    printf("%d",(*q)[1]);
}
int main()
{
    int a[][2]={2,4,6,8,10,12};
    int (*ptr)[2]=a;
    f(ptr);
    return 0;
}
```

The output is:

- | | |
|----------|---------|
| (a) 4 12 | (b) 4 8 |
| (c) 2 10 | (d) 2 6 |

[MCQ]

7. Consider the following program:

```
#include<stdio.h>
int main()
{
    int a[3]={0,1,2};
    int *p=(int *)(&a+1);
    printf("%d\t%d", *(a+1),*(p-1));
    return 0;
}
```

The output is-

- (a) Garbage value
- (b) Segmentation fault
- (c) 1 2
- (d) Compilation Error

[MCQ]

8. Consider the following program:

```
#include<stdio.h>
void fun(int n){
    for(n--;--n;--n)
        printf("GATE WALLAH");
}
int main()
{
    void (*p)(int)=fun;
    (*p)(6);
    return 0;
}
```

The output is-

- (a) Compilation Error
- (b) Runtime Error
- (c) printf() is executed infinite number of times.
- (d) print() is executed two times.

Answer Key

- 1. (21)
- 2. (c)
- 3. (382)
- 4. (b)
- 5. (d)

- 6. (a)
- 7. (c)
- 8. (d)



Hints and solutions

1. (21)

	1000
p	10 20 21
	2000

q	1000
---	------

$*q=p+++*q;=10+*1000=10+10=20;$
p is then incremented by 1.

Final value at p=21

2. (c)

No error, since it returns the address of static array.

for($i=0;i<3;i++$) printf("%d\t", p[i]+p[i+1]);

The loop prints $p[0]+p[0+1]$, $p[1]+p[1+1]$,

$p[2]+p[2+1]$.

Output: 3 5 7

3. (382)

	1000	3000
p	10 20 21	s 20 19 361
	2000	4000

q	1000 3000	r 2000
---	-----------	--------

$$p+s = 21 + 361 = 382$$

4. (b)

Runtime error exists since it returns the address of local array variable.

5. (d)

Both the statements P and Q are CORRECT.

P: CORRECT.

int * p(int *) - p is a function that takes an integer pointer as argument and returns an integer pointer.

Q: CORRECT

int (*p(int *))[] - p is a function that takes an integer pointer as argument and returns a pointer to an array of integers.

6. (a)

```
void f(int (*q)[2]){
    printf("%d\t",(*q)[1]); //It prints the first element of the 0th row of a. So, 4 is printed.
    q+=2; //q now points to the 2nd row of a.
    printf("%d",(*q)[1]); // It prints the first element of the 2nd row of a. So, 12 is printed.
}
```

Output: 4 12

7. (c)

Suppose the elements 0, 1, 2 are stored at locations 100, 102, 104 (assuming integer size of 2 bytes).

```
int a[3]={0,1,2};
int *p=(int *)(&a+1); //p contains 106
printf("%d%d", *(a+1),*(p-1)); //*(100+1)=*102=1 and *(106-1)=*104=2. So, 1 2 are printed.
return 0;
}
Output: 1 2
```

8. (d)

p is a pointer to the function fun.

```
void fun(int n){ //n=6
    for(n--; --n ;--n)
        6 4 -> printf() 3
        2 -> printf() 1
    0->Loop stops
}
```

So, printf() is executed 2 times.



Any issue with DPP, please report by clicking here:- <https://forms.gle/t2SzQVvQcs638c4r5>

For more questions, kindly visit the library section: Link for web: <https://smart.link/sdfez8ejd80if>



PW Mobile APP: <https://smart.link/7wwosivoicgd4>



Programming in C

Arrays and Pointers

DPP-04

[MCQ]

1. Consider the following codes:

```
P: void *p;
p=malloc(1);
*p=65;
printf("%c",*(char*)p);
```

```
Q: void *p;
char a='A';
p=malloc(1);
p=&a;
printf("%c",*(char*)p);
```

Which of the following is CORRECT?

- (a) Both P and Q are valid.
- (b) Only P is valid.
- (c) Only Q is valid.
- (d) Neither P nor Q is valid.

[MSQ]

```
2. #include <stdio.h>
#include <stdlib.h>
int * f()
{
    int *p=(int*)malloc(sizeof(int));
    *p=10;
    return p;
}
int * g(int a)
{
    return &a;
}
int main()
{
    printf("%p", f());//line 1
    printf("%p", g(15));//line 2
    return 0;
}
```

Which of the following statement(s) is/are INCORRECT?

- (a) Line 1 will result into compilation error.
- (b) Line 2 will result into compilation error.
- (c) The outputs are garbage values.

- (d) The hexadecimal addresses of pointer variables p and local variable are displayed.

[MCQ]

3. #include <stdio.h>

```
int main()
{
    void *p, *q;
    int a=324;
    p=&a;
    printf("%d", *(char*)p);
    return 0;
}
```

The output is-

- (a) Garbage value
- (b) Compilation error
- (c) 68
- (d) 324

[NAT]

4. #include <stdio.h>

```
#include <stdlib.h>
int main()
{
    int *p=(int*)malloc(sizeof(int));
    int *q=(int*)malloc(sizeof(int));
    *p=376;
    *q=5;
    while(*p>*q){
        printf("%d\t",*p);
        *p-=*q;
        *q+=1;
    }
    return 0;
}
```

The sum of the printed values is _____.

[MCQ]

5. #include <stdio.h>
- #include <stdlib.h>
- int main() {

```

int count=0;
char *p=(char *)malloc(sizeof(char));
*p=65;
printf("%c",*p);
p=realloc(p, 4*sizeof(char));
*p=256;
printf("%d",*(int*)p);
return 0;
}

```

The output printed is-

- (a) A followed by Garbage values
- (b) A0
- (c) A512
- (d) Compilation error

[MCQ]

6. #include <stdio.h>

```

#include <stdlib.h>
int * f()
{
    int *p=(int*)malloc(sizeof(int));
    *p=20;
    return p;
}
int * g()
{
    static int a=10;
    int *q;
    q=&a;
    return q;
}
int main()
{
    printf("%d\t", *g());//line 1
    printf("%d", *f());//line 2
    return 0;
}

```

The output is-

- (a) Garbage value
- (b) Compilation error
- (c) 10 20
- (d) 20 10

[MCQ]

7. When the memory is full, malloc returns-

- (a) Void pointer

- (b) Wild pointer
- (c) Dangling pointer
- (d) NULL pointer

[MCQ]

8. #include <stdio.h>

```

#include <stdlib.h>
int main()
{
    int *p=(int *)calloc(2, sizeof(int));
    int *q;
    q=p+1;
    printf("%d\t", *p);
    printf("%d\t", *q);
    *p=10;
    *q=15;
    printf("%d\t", *p);
    printf("%d\t", *q);
    free(p);
    return 0;
}

```

The output is:

- (a) 10 15 Garbage 15
- (b) Garbage Garbage 10 15
- (c) 0 0 10 15
- (d) 10 15 0 0

Answer Key

- | | |
|--------------|--------|
| 1. (c) | 5. (b) |
| 2. (a, c, d) | 6. (c) |
| 3. (c) | 7. (d) |
| 4. (463) | 8. (c) |



Hints and solutions

1. (c)

P: void *p;
 $p=\text{malloc}(1);$
 $*p=65;$ //Invalid use of void expression. Void pointer stores the address of any variable and needs proper typecasting.
 $\text{printf}("%c",*(\text{char}*)p);$
Hence, P is **invalid**.

Q: void *p;

char a='A';
 $p=\text{malloc}(1);$
 $p=&a;$ //Void pointer p is storing the address of char variable 'a'.
 $\text{printf}("%c",*(\text{char}*)p);$ //Proper type casting is done hence, the code Q is **valid**.

2. (a, c, d)

%p is the format specifier for hexadecimal memory address.

The function g() is returning the address of local variable 'a' which will go out of scope as soon as g() finishes execution.

So, Line 2 in the main() given as-
 $\text{printf}("%p", g(15));$
will give ERROR.

3. (c)

The binary of 324 is $(101000100)_2$.

The address of the variable a containing 324 is stored in void pointer p.

It is type-casted to char* i.e. it dereferences only 8 bits from right i.e $(1000100)_2$. Its decimal equivalent is 68.
Output: 68

4. (463)

$*p=376$
 $*q=5$
 $\text{while}(376>5)\rightarrow\text{True}$
 $*p=376\rightarrow\text{printf}()$ executed.
 $*p=376/5=75$
 $*q=5+1=6$

$\text{while}(75>6)\rightarrow\text{True}$

$*p=75\rightarrow\text{printf}()$ executed.

$*p=75/6=12$

$*q=6+1=7$

$\text{while}(12>7)\rightarrow\text{True}$

$*p=12\rightarrow\text{printf}()$ executed.

$*p=12/7=1$

$*q=7+1=8$

$\text{while}(1>8)\rightarrow\text{False}$

STOP.

Output: 376 75 12

Sum: 463

5. (b)

```
#include <stdio.h>
#include <stdlib.h>
int main() {
    char *p=(char *)malloc(sizeof(char));
    *p=65;
    printf("%c",*p); //A
    p=realloc(p, 4*sizeof(char)); //realloc reallocates the
    memory with size equivalent to 4 bytes and stores
    the address in pointer variable 'p'.
    *p=256;
    printf("%d",*(int*)p); // This will result into
    overflow.
    return 0;
}
```

Output: A0

6. (c)

A static variable has scope throughout the program.

The function g() is returning the address of static variable 'a'.

$\text{printf}("%d\t", *g());$ //10

$\text{printf}("%d", *f());$ //20

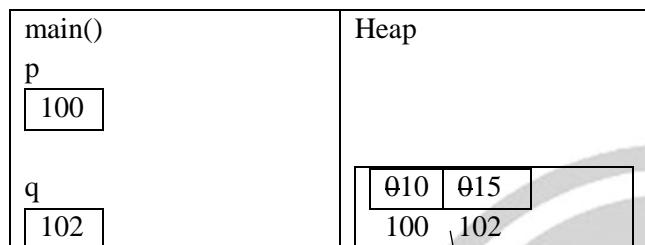
Output: 10 20

7. (d)

When the memory is full, malloc() returns NULL pointer.

8. (c)

calloc() allocates two continuous memory block of the sizes equivalent to store an integer. It returns the address to integer pointer p.



```
printf("%d\t", *p);//0
printf("%d\t", *q);//10
*p=10;
*q=15;
printf("%d\t", *p);
//10
printf("%d\t", *q);
//15
free(p); //free(100)
//Entire block allocated
by calloc() is freed.
```

Free(100) frees the entire space.

Output: 0 0 10 15



Any issue with DPP, please report by clicking here:- <https://forms.gle/t2SzQVvQcs638c4r5>

For more questions, kindly visit the library section: Link for web: <https://smart.link/sdfez8ejd80if>



PW Mobile APP: <https://smart.link/7wwosivoicgd4>

C Programming

Control Flow Statement



DPP-01

[NAT]

```
1. #include <stdio.h>
int main(void){
    int i = 2, j = 3, k = 4;
    if (i < j ? 1 : 0)
        printf("GATE");
    else
        printf("Wallah2023");
    return 0;
}
```

The output of the program is _____.

[MCQ]

```
2. #include <stdio.h>
void main( ){
    int a, b, c, d;
    a = 2; b = -1; c = 3; d = -4;
    if(a = b - c - d)
        printf("%d%d%d", a++, b--, c++);
    else
        printf("%d%d%d", c--, ++a, ++b);
}
```

The output is _____.

(a) 1 -2 4
 (b) 3 1 0
 (c) 2 1 -3
 (d) 3 3 0

[MCQ]

```
3. #include <stdio.h>
int main(void){
    int a = 3 > 2 ? 0 : 0 : 1 : 5;
    if(a == a - 1)
        printf("GATE 2023");
    else
        printf("GATE WALLAH");
    return 0;
}
```

The output of the program is _____.

(a) GATE 2023

- (b) GATE WALLAH
- (c) Compiler error
- (d) Garbage value

[NAT]

```
4. #include <stdio.h>
void main( ){
    int a;
    a = printf("GATE Wallah 2023");
    if(a%4 == 0)
        a = a + 5;
    else
        a = a - 5;
    printf("%d", a++);
}
```

The value of a at the end of the program is ____.

[NAT]

```
5. #include <stdio.h>
void main( ){
    int i, j, k;
    j = 4;
    k = 0;
    i = j < k ? k : j--;
    if(j < i)
        j = j + k - 1;
    if(j == i)
        j = j - i;
    else
        j = j + --k;
    printf("%d", j + k - i);
}
```

The output is ____.

[NAT]

6. Consider the following program:

```
#include<stdio.h>
int main()
{
    int a=19, b=20;
```

```

if(a++<b--) printf("%d",a++++-b);
else printf("%d", ++a--b);
return 0;
}

```

The output is _____.

[MCQ]

```

7. #include<stdio.h>
void main()
{
int a=0;
printf("%d", a);
if(a==2){
    printf("Hi");
    printf("%d",a);
}else{
    printf("Bye");
}
printf("%d", a);

```

}

The output string is:

- | | |
|-----------|-----------|
| (a) 0Hi22 | (b) 0Hi20 |
| (c) 0Bye0 | (d) 0Hi00 |

[MCQ]

```

8. #include<stdio.h>
void main()
{
int a=0, b=0;
a=(a==4)||(b==1);
if(a&&b) printf("CProgramming");
else printf("PankajSharma");
printf("%d",b);
}

```

The output is-

- | |
|-------------------|
| (a) CProgramming0 |
| (b) CProgramming1 |
| (c) PankajSharma0 |
| (d) PankajSharma1 |

Answer Key

- 1. (GATE)
- 2. (b)
- 3. (b)
- 4. (22)
- 5. (-4)

- 6. (38)
- 7. (a)
- 8. (c)



Hints and solutions

1. (GATE)

$i < j ? 1 : 0$

In the above expression i value is less than j value, hence it will return 1.

So, it will print GATE.

2. (c)

$a = -1 - 3 + 4$

$a = 0$

$a \boxed{2} 0$

Assignment operator assigns and returns the value

$b \boxed{++b: -1} 0 ,$

$a \boxed{++a: \emptyset} 1 ,$

$c \boxed{c--: \not\emptyset} 2$

\Downarrow

Post decrement (It will print 3, then update to 2)

Output: 3 1 0

3. (b)

$a = \frac{3 > 2}{\downarrow} ? \boxed{0 ? 0 : 1} : 5$

$a = 1$

Assignment operator assigns the value and returns it

if $\underbrace{(a = a - 1)}_0$

\Downarrow
Condition: false

Output: GATE WALLAH

4. (GATE Wallah 202321)

GATE Wallah 2023

\Downarrow

$a = 16$

printf returns the number of characters successfully printed

$16 \% 4 = 0 \rightarrow \text{True}$

\Downarrow

$a = a + 5$

$a \boxed{16 21} 22$

Hence the final value of a is 22.

5. (-4)

$i \quad j \quad k$
 $4 \not\mid \emptyset -1$

$\not\mid \not\mid 1$

$3 < 4$

$j = 3 + 0 - 1 = 2$

$2 \neq 4$

$j = j - 1$

$= 2 - 1$

$j = 1$

printed value = $j + k - i$
 $= 1 - 1 - 4$
 $= -4$

6. (38)

If $(19 < 20) \rightarrow$ Condition is true. After the condition is evaluated, a is incremented to 20 and b is decremented to 19.

Now, $\text{printf}("%d", a++++-b);$ is evaluated. b is decremented to 18. So, $(20+18)$ i.e. 38 is printed. After that, a is incremented to 21.

Hence, output is 38.

7. (a)

void main()

{

int a=0;

printf("%d", a); // 0 is printed

if($a=2$) { // Assignment operator assigns and returns the assigned value; So 2 is assigned to a and 2 is returned. Any non-zero value is considered true.

printf("Hi"); // "Hi" is printed

```

printf("%d",a);//Since a contains 2, 2 is printed.
}else{
    printf("Bye");
}
printf("%d", a); //Since a contains 2, 2 is printed
}
Output: 0Hi22

```

8. (c)

a=0. b=0;

$a=(a=4)\|(b=1)$ //Assignment operator assigns and returns the assigned value. Here, short-circuiting will be applied. Since the logical operator is OR, if the first part is true, second part is not evaluated at all.

Hence, b=0, a=1.

if(a && b)//The condition evaluates to 1 && 0 i.e. 0.
Hence, else part is evaluated.

Output: PankajSharma0



Any issue with DPP, please report by clicking here:- <https://forms.gle/t2SzQVvQcs638c4r5>

For more questions, kindly visit the library section: Link for web: <https://smart.link/sdfez8ejd80if>



PW Mobile APP: <https://smart.link/7wwosivoicgd4>

C Programming

Control Flow Statement



DPP-02

[NAT]

1. Consider the following program:

```
#include<stdio.h>
int main()
{
    int i=3;
    for(;i++<=9;++i);
    printf("%d",i);
    return 0;
}
```

The output is _____.

[NAT]

2. Consider the following program:

```
#include<stdio.h>
int main()
{
    int i=3;
    for(;i++<=9;++i)
    {
        printf("best hai GATE Wallah");
        if(i>8) break;
    }
    return 0;
}
```

The number of times printf() executed is _____

[MCQ]

3. Consider the following program:

```
#include<stdio.h>
int main()
{
    int a=0, b=1;
    for(;b;printf("%d\t", a))
    {
        b=a++<=3;
    }
    return 0;
}
```

}

The output sequence is-

- | | |
|---------------|---------------|
| (a) 0 1 2 3 4 | (b) 0 1 2 3 |
| (c) 1 2 3 4 | (d) 1 2 3 4 5 |

[MCQ]

4. Consider the following program:

```
#include<stdio.h>
int main()
{
    int i=0;
    for(i=2; i<=n; i*=2)
    printf("GATE WALLAH 2024");
    return 0;
}
```

The number of times printf() executed is approximated as-

- | | |
|-----------------------|----------------|
| (a) 2^n | (b) $\log_2 n$ |
| (c) $\log_2 \log_2 n$ | (d) 2 |

[MCQ]

5. Consider the following program:

```
#include <stdio.h>
int main()
{
    int i;
    for(i=printf("GATE");i<printf("Wallah2023");
    {
        if(i%2==0) i=i-1;
        else i=i+2;
    }
    return 0;
}
```

The number of times "Wallah2023" is printed is-

- | | |
|-------|-------|
| (a) 4 | (b) 5 |
| (c) 6 | (d) 7 |

[MCQ]

6. Consider the following program:

```
#include <stdio.h>
int main()
{
    int i, j, n;
    for(i=1;i<=n;i++)
        for(j=1;j<=n;j++)
            printf("GATE Wallah");
    return 0;
}
```

How many times will the printf() statement be executed when n=32?

- (a) 128
- (b) 1024
- (c) 512
- (d) 256

[NAT]

7. Consider the following program:

```
#include <stdio.h>
int main()
{
    int i, j, n, count=0;
    for(i=1;i<=n;i++)
        for(j=1;j<=i;j++)
            count+=1;
```

```
printf("%d", count);
return 0;
}
```

When n=31, the output is _____

[MCQ]

8. Consider the following program:

```
#include <stdio.h>
int main()
{
    int a,i;
    a= 12.5/5+31.2/2*5-5;
    for(i=a; i<90;i=i+3)
    {
        printf("%c\t", i+32);
    }
    return 0;
}
```

The output is-

- (a) 107 110 113 116 119
- (b) K L M N O
- (c) k n q t w
- (d) Garbage values

Answer Key

- 1. (12)
- 2. (4)
- 3. (d)
- 4. (b)
- 5. (c)

- 6. (b)
- 7. (496)
- 8. (c)



Hints and solutions

1. (12)

$3 \leq 9$ TRUE. i is incremented to 4. Loop is executed.
 ++i is executed. i takes value 5.
 $5 \leq 9$ TRUE. i is incremented to 6. Loop is executed.
 ++i is executed. i takes value 7.
 $7 \leq 9$ TRUE. i is incremented to 8. Loop is executed.
 ++i is executed. i takes value 9.
 $9 \leq 9$ TRUE. i is incremented to 10. Loop is executed.
 ++i is executed. i takes value 11.
 $11 \leq 9$ FALSE. i is incremented to 12. Loop is not executed.

Final value of i= 12.

2. (4)

$3 \leq 9$ TRUE. i is incremented to 4 as $i++$ was used.
 printf() executed.
 $(i > 8)$ false;
 ++i is executed. i takes value 5.
 $5 \leq 9$ TRUE. i is incremented to 6 as $i++$ was used.
 printf() executed.
 $(i > 8)$ false;
 ++i is executed. i takes value 7.
 $7 \leq 9$ TRUE. i is incremented to 8 as $i++$ was used.
 printf() executed.
 $(i > 8)$ false;
 ++i is executed. i takes value 9.
 $9 \leq 9$ TRUE. i is incremented to 10 as $i++$ was used.
 printf() executed.
 $(i > 8)$ True; So, it breaks out from the loop.

Therefore, printf() is executed 4 times.

3. (d)

Condition of for loop: b
 $1 \rightarrow$ TRUE; $b = a++ \leq 3$ is executed; $b = 0 \leq 3 \rightarrow$ TRUE;
 $b = 1$, a is incremented to 1.
1 is printed.
 $1 \rightarrow$ TRUE; $b = a++ \leq 3$ is executed; $b = 1 \leq 3 \rightarrow$ TRUE;
 $b = 1$, a is incremented to 2.
2 is printed.
 $1 \rightarrow$ TRUE; $b = a++ \leq 3$ is executed; $b = 2 \leq 3 \rightarrow$ TRUE;
 $b = 1$, a is incremented to 3.

3 is printed.

$1 \rightarrow$ TRUE; $b = a++ \leq 3$ is executed; $b = 3 \leq 3 \rightarrow$ TRUE;
 $b = 1$, a is incremented to 4.
4 is printed.
 $1 \rightarrow$ TRUE; $b = a++ \leq 3$ is executed; $b = 4 \leq 3 \rightarrow$ FALSE;
 $b = 0$, a is incremented to 5.
5 is printed.
 $0 \rightarrow$ FALSE; Loop terminates.
Output: 1 2 3 4 5

4. (b)

The for loop runs for i values- $2, 2^2, 2^3 \dots 2^k$

The loop terminates when $2^{k+1} > n$

$$k = \log_2 n$$

5. (c)

$i = \text{printf}("GATE"); // i is initialized to 4$
 $i < \text{printf}("Wallah2023"); // i < 10$
 $\downarrow \text{printf}() \text{ executed}$

i 4 3 5 7 9

i takes values →

$$(4 \% 2) == 0 \rightarrow i = i - 1 \Rightarrow i = 3$$

$3 < 10 \rightarrow \text{printf}()$ executed

$$(3 \% 2)! = 0 \rightarrow i = i + 2 = 3 + 2 = 5$$

$5 < 10 \rightarrow \text{printf}()$ executed

$$(5 \% 2)! = 0 \rightarrow i = 7$$

$7 < 10 \rightarrow \text{printf}()$ executed

$$(7 \% 2)! = 0 \rightarrow i = 9$$

$9 < 10 \rightarrow \text{printf}()$ executed

$$(9 \% 2)! = 0 \rightarrow i = 11$$

$11 < 10 \rightarrow \text{printf}()$ executed but loop terminates as condition is false.

$\therefore \text{"Wallah2023"} \text{ is printed 6 times.}$

6. (b)

The given nested loops are independent.

The loop will run $(n*n)$ times.

$\therefore \text{The printf() statement is executed for } (32*32) = 1024 \text{ times when } n = 32.$

7. (496)

The given nested loops are dependent loops.

i	1	2	3	n
j	1	1, 2	1, 2, 3		1, 2, 3, ..., n
Count	1	(1+2)	(1+2+3)		(1+2+3+... n)

The final value of count is—

$$\Rightarrow (1 + 2 + 3 + \dots + n)$$

$$\Rightarrow (1 + 2 + 3 + \dots + 31) \text{ when } n = 31$$

$$\Rightarrow \frac{31 \times 32}{2}$$

$$\Rightarrow 496$$

8. (c)

$$a = 12.5/5 + 31.2/2 * 5 - 5$$

$$a = 75.5$$

$\because a$ is integer

$$\therefore a = 75$$

The for loop converts each character to lower case corresponding to the ASCII values.

Output ASCII : $(75 + 32) (78 + 32) \dots (87 + 32)$

Output characters : k n q t w



Any issue with DPP, please report by clicking here:- <https://forms.gle/t2SzQVvQcs638c4r5>

For more questions, kindly visit the library section: Link for web: <https://smart.link/sdfez8ejd80if>



PW Mobile APP: <https://smart.link/7wwosivoicgd4>

C Programming

Control Flow Statement



DPP-03

[NAT]

```
1. #include <stdio.h>
int main()
{
    int a=7, b=8;
    while(++b & a-- )
    {
        printf("HI!");
    }
    return 0;
}
```

The number of times the printf() executed is _____.

[MCQ]

```
2. #include <stdio.h>
int main()
{
    int a=5, b=10;
    while(++b & a-- )
    {
        switch (b)
        {
            case 0: b=b-1;
            break;
            case 1: b=b-2;
            break;
            default: b=b-3;
            break;
        }
    }
}
```

```
}
printf("%d\t%d", a, b);
return 0;
}
```

The output is-

- | | |
|----------|---------|
| (a) 4 8 | (b) 3 9 |
| (c) 3 10 | (d) 4 6 |

[MCQ]

```
3. int main()
{
    int a=1, b=2;
    do
    {
        while(b++)
        {
            b=b-a;
            a=a+b;
        }
    }
    while(a++<2);
    printf("%d\t%d", a, b);
    return 0;
}
```

The output is-

- | | |
|---------|---------|
| (a) 3 0 | (b) 4 2 |
| (c) 3 2 | (d) 4 1 |

[MCQ]

4. #include <stdio.h>

```
int main()
{
    int a,b;
    a=printf("GATE")>printf("Wallah")?printf("2023"):printf("Hi!!");
    b=a-1;
    while(a>b)
    {
        switch(b)
        {
            case 1: b=--a;
            case 2:b=a--;
            default:b=--a;
        }
    }
    printf("%d%d", a, b);
    return 0;
}
```

The output string is-

- (a) GATEWallahHi!!33
- (b) GATEWallah202303
- (c) GATEWallah202330
- (d) GATEWallahHi!!00

[NAT]

5. #include <stdio.h>

```
int main()
{
    int i=16;
    do
    {
        i=i-2;
    }
```

```
    printf("%d",i);
}
while(i++);
return 0;
}
```

The sum of all printed values of i is ____?

[MCQ]

6. #include <stdio.h>

```
int main()
{
    int a, b;
    while(a!=b)
    {
        a=a/2;
        b=b*2;
        if(b>a) break;
    }
    return 0;
}
```

If $a = 2^m$ and $b = 2^n$ where $m-n$ is even and positive, the number of times the loop runs is-

- | | |
|---------------------|--|
| (a) $\frac{m-n}{2}$ | (b) $\left\lceil \frac{m-n}{2} \right\rceil + 1$ |
| (c) $\frac{n-m}{2}$ | (d) $\left\lceil \frac{n-m}{2} \right\rceil + 1$ |

[NAT]

7. #include <stdio.h>

```
int main()
{
    int x=5, y=10;
    if(printf("GATE")-3){
        while(x--) y=y+x;
    } else y=y-x;
}
```

```
return 0;
}
```

The value of y at the end of the program is _____.

[NAT]

8. #include <stdio.h>
 int main()
 {
 int x=5, y=5;

```
while(x-=y++<10){  

printf("GATE WALLAH\n");  

}  

return 0;  

}
```

The number of times “GATE WALLAH” printed is
 _____.



Answer Key

- 1. (7)
- 2. (b)
- 3. (d)
- 4. (a)
- 5. (105)

- 6. (a)
- 7. (20)
- 8. (4)

Hints and solutions

1. (7)

a = 7 = 0111 a $\boxed{1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0}$
 $\boxed{1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0}$
 $\boxed{1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0}$ ∴
 b $\boxed{1 \ 0 \ 0 \ 0}$ b $\boxed{1 \ 0 \ 0 \ 0}$

while (++b & a --)

9 & 7 → true 1001
 0111
 0001
 printf → executed

10 & 6 → True. 1010
 0110
 0010
 printf → executed ;

similar explanation –

∴ printf() is executed –

for a = 7, 6, 5, 4, 3, 2, 1 → Total 7 times

2. (b)

a $\boxed{1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0}$ a $\boxed{1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0}$
 b $\boxed{1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0}$ b $\boxed{1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0}$
 while (++b & a --)
 11 & 5 → true
 switch (11)
 default: b = 11 - 3

9 & 4 → false

∴ Final values of a and b are –

$\boxed{3 \ 9}$

3. (d)

a $\boxed{1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0}$ a $\boxed{1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0}$
 b $\boxed{1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0}$ b $\boxed{1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0}$
 do
 {
 While (b++)
 {
 // values assigned are 2, 2, 0
 b = b - a = 3 - 1 = 2 // 3 - 3 = 0
 a = a + b = 1 + 2 = 3 // 3 + 0 = 3
 }
 }
 }

while (a ++ < 2);

3 < 2 → false

final values are –

$\begin{array}{c} a | b \\ 4 | 1 \end{array}$

4. (a)

a = print f (“GATE”) > printf (“Wallah”)?printf(“2023”): printf(“Hi!!”);
 ↓ > 6 → False

a = 4; Output:

b = a-1; // b = 3 GATEWallahHi!!33

while (a > b){

4 > 3

3 > 3 → false

switch (b){

3

case 1 : b = -- a;

case 2 : b = a --;

a $\boxed{43}$

default : b = -- a;

b $\boxed{33}$

}

}

5. (105)

Do

{

i = i - 2;

printf(“%d”,j); //14 13 0

}

while (i ++);

14

13

.

.

$0 \rightarrow$ stop.

Value printed—

14 13 12 11 1 0

Sum of the values—

$$\Rightarrow \frac{14 \times (14 + 1)}{2}$$

$$\Rightarrow 105$$

6. (a)

$$a = 1024; \quad b = 64$$

while ($a \neq b$)

{

$1024 \neq 64 \rightarrow$ True

$256 \neq 256 \rightarrow$ False

$$a = a/2; //a = 512, 256$$

$b = b * 2; //b = 128, 256$

if ($b > a$) break;

}

\therefore The loop runs for 2 times $\left(\because \frac{10 - 6}{2} = 2\right)$

7. (20)

printf() prints and returns the number of characters it successfully printed. So, the condition becomes (4-3) i.e 1 which is TRUE.

So, the while loop is executed for x values= 5, 4, 3, 2, 1

$$y = 10 + 4 + 3 + 2 + 1 + 0 = 20$$

8. (4)

$x = y++ < 10$ will make the condition true for $x = 4, 3, 2, 1$ and $y = 5, 6, 7, 8$.

Hence, printf() will be executed 4 times.



Any issue with DPP, please report by clicking here:- <https://forms.gle/t2SzQVvQcs638c4r5>

For more questions, kindly visit the library section: Link for web: <https://smart.link/sdfez8ejd80if>



PW Mobile APP: <https://smart.link/7wwosivoicgd4>

C Programming

Control Flow Statement



DPP-04

[NAT]

- ```
1. #include <stdio.h>
void main()
{
 int i = 0;
 switch(i)
 {
 case 0: i = i + 1;
 case 1: i = i + 3;
 case 2: i = i * 2;
 break;
 default: i = i + 5
 }
}
```

The value of i is \_\_\_\_\_.

[NAT]

- ```

2. #include <stdio.h>
void main( )
{
    int a, b, c, d, e;
    b = 1; c = 1; d = 2; e = -1;
    a = b++ && c-- || d++ && e--;
    switch(c)
    {
        case 0: a = a + 1;
        case 1: a = a - 1;
        default: a = a - 2;
    }
}

```

The final value of $(a + b + c + d + e)$ is

[MCQ]

- ```
3. #include <stdio.h>
int main(void)
{
 int x ;
 scanf("%d", &x);
```

`switch(x)`

```

 {
 case 0: x = x + 1;
 break;
 default: x = x - 1;
 case 1: x = x - 11;
 case 2: x = x + 128;
 break;
 }
 printf("%c", x);
 return 0;
}

```

What is the output when  $x = -4$ ?



**[MCQ]**

- ```
4. #include <stdio.h>
int main(void)
{
    int q, r = 0;
    q = 2 * 3/6 + 2.0/5 + 0.2 * 3;
    r = -q--;
    switch(q - r)
    {
        case 0: printf("Hello");break;
        case 1: printf("Hi");break;
        case 2: printf("best hai");break;
        case 3: printf("GATE Wallah");break;
        default: printf("2023");
    }
}
```

The output of the program is

- (a) Hibest haiGATE Wallah
 - (b) best haiGATE Wallah
 - (c) GATE Wallah2023

(d) GATE Wallah

[MCQ]

5. #include<stdio.h>
 int main()
 {
 int j=10, p=0;
 for(j>0;)
 {
 switch(j)
 {
 case 1: p+=3;
 case 2: p+=5;
 break;
 default: p-=8;
 case 3: p-=7;
 break;
 }
 j=j-3;
 printf("%d\t",p);
 }
 return 0;
 }

The output is-

- (a) -15 -30 -42 -45
- (b) -15 -45 -30 -42
- (c) -15 -30 -42 -39
- (d) -15 -30 -45 -37

[NAT]

6. #include<stdio.h>
 int main()
 {
 int x;
 for(x=0; x<32; x++)
 {
 switch(x)

```
{
    case 0: x= x+2;
    case 1: x=x+5;
    case 2: x=x+1;
    default: x=x+7;
}
printf("%d\t",x);
}
return 0;
}
```

The sum of the values printed is _____.

[MCQ]

7. . Consider the following two statements:
 P: Case label can be integer or character or floating point numbers.
 Q: Only one default is allowed in switch-case structure.
 Which of the following statements are INCORRECT?
 (a) Both P and Q (b) Only P
 (c) only Q (d) Neither P or Q.

[MCQ]

8. #include<stdio.h>
 int main()
 {
 int x=4, y=5;
 x=x==y==5;
 switch(1)
 {
 x=x+11;
 }
 printf("%d", ++x);
 return 0;
}

The output is-

- (a) 0 (b) 1
- (c) 11 (d) Compiler Error

Answer Key

- 1. (8)
- 2. (2)
- 3. (a)
- 4. (c)
- 5. (d)

- 6. (69)
- 7. (b)
- 8. (b)



Hints and solutions

1. (8)

If no break statements exist then all the case statements are executed

```
i = 0;
i = 0 + 1 = 1;
i = 1 + 3 = 4;
i = 4 * 2 = 8
```

2. (2)

b $\boxed{\cancel{1} 2}$	c $\boxed{\cancel{1} 0}$	d $\boxed{2}$	e $\boxed{-1}$
a $\boxed{-1}$			
a = b++ && c-- d++ && e--			

This part won't be evaluated because of short circuit

```
(1 && 1)
↓
True
switch(0)
{
    case 0: a = a + 1 ⇒ a = 1 + 1 = 2
    case 1: a = a - 1 ⇒ a = 2 - 1 = 1
    default: a = a - 2 ⇒ a = 1 - 2 = -1
    ∴ a + b + c + d + e = -1 + 0 + 2 - 1 = 2
}
```

3. (a)

X = -4, default case is executed. Since there are no breaks, case 1 and case 2 will also be executed.

```
x=x-1;//x=-5
x=x-11;//x=-16
x=x+12;//x=112
```

The equivalent character with ASCII value 112 is p.

4. (c)

$q = 1 + 0.4 + 0.6 = 2.0$ when assigned to integer variable, q=2

$$\begin{array}{r} q \boxed{\cancel{1}} \\ r \boxed{\emptyset - 2} \\ r = r - q \end{array}$$

After this q is decremented to 1.

$$q - r = 1 + 2 = 3$$

Output: GATE Wallah2023

[Note: there is no break after case 3]

5. (d)

```
j=10;
switch(10)
{
    case 1: p+=3;
    case 2: p+=5;
    break;
    default: p-=8;//p= 0 - 8 = -8
    case 3: p-=7;//p= -8 - 7 = -15
    break;
}
j=j-3;//j=7
printf("%d\t",p); // -15
j=7;
switch(7)
{
    case 1: p+=3;
    case 2: p+=5;
    break;
    default: p-=8;//p= -15 - 8 = -23
    case 3: p-=7;//p= -23 - 7 = -30
    break;
}
j=j-3;//j=4
printf("%d\t",p); // -30
j=4;
switch(4)
{
    case 1: p+=3;
    case 2: p+=5;
    break;
```

```

default: p-=8;//p= -30 - 8= -38
case 3: p=7;//p= -38 -7 = -45
break;
}
j=j-3;//j=1
printf("%d\t",p); // -45
j=1;
switch(1)
{
    case 1: p+=3; //p= -45 +3 =-42
    case 2: p+=5; //p = -42 + 5 =-37
    break;
    default: p=8;
    case 3: p=7;
    break;
}
j=j-3;//j=-2
printf("%d\t",p); // -37
Output: -15 -30 -45 -37

```

6. (69)

```

x=0; 0<32 -> TRUE
switch(0){
    case 0: x= x+2;//x=0+2=2
    case 1: x=x+5;//x=2+5=7
    case 2: x=x+1;//x=7+1=8
    default: x=x+7;//x=8+7=15
}
printf("%d\t",x);//15 is printed
x is incremented to 16.
x=16; 16<32-> TRUE
switch(16)
{
    case 0: x= x+2;
    case 1: x=x+5;

```

```

        case 2: x=x+1;
        default: x=x+7;//x=16+7=23
    }
    printf("%d\t",x);//23 is printed
    x is incremented to 24.
    x=24; 24<32-> TRUE
    switch(24)
    {
        case 0: x= x+2;
        case 1: x=x+5;
        case 2: x=x+1;
        default: x=x+7;//x=24+7=31
    }
    printf("%d\t",x);//31 is printed
    x is incremented to 32.
    32<32 is FALSE. Execution stops.
    Sum of printed values= 15+23+31=69

```

7. (b)

P: INCORRECT. Case label can never be floating point numbers.

Q: CORRECT. Only one default is allowed in switch-case structure.

8. (b)

```

x=4, y=5
x==y==5;
x==y is 0 and 0==5 is 0.
So x==y==5 is equivalent to x=0.
The switch is never executed here.
So, printf("%d", ++x) increments x to 1 and prints it.
Output: 1

```



Any issue with DPP, please report by clicking here:- <https://forms.gle/t2SzQVvQcs638c4r5>

For more questions, kindly visit the library section: Link for web: <https://smart.link/sdfez8ejd80if>



PW Mobile APP: <https://smart.link/7wwosivoicgd4>

C Programming

Data Types & Operators

DPP-01

[MCQ]

1. Consider the following declarations:

P: signed short x;

Q: unsigned long long int x;

Which of the given declarations is/are CORRECT?

- (a) Only P
- (b) Only Q
- (c) Both P and Q
- (d) Neither P nor Q

[NAT]

2. Consider the following program:

```
#include <stdio.h>
int main()
{
    int x=32769;
    printf("%d", x);
    return 0;
}
```

(Assume integer is of 2 bytes)

The value printed is- _____.

[MCQ]

3. Consider the following program:

```
#include <stdio.h>
int main()
{
    char ch=141;
    printf("%d", ch);
    return 0;
}
```

The output is-

- (a) Compiler Error
- (b) -115
- (c) -128
- (d) 141

[MCQ]

4. Consider the following function:

```
#include <stdio.h>
int main()
{
    char ch = -134;
    printf("%c", ch);
    return 0;
}
```

The output is-

- (a) A
- (b) Garbage
- (c) Compiler Error
- (d) z

[NAT]

5. Consider the following program:

```
#include <stdio.h>
int main()
{
    char ch=125;
    ch=ch+6;
    printf("%d", ch);
    return 0;
}
```

The output is- _____.

[NAT]

6. Consider the following program:

```
#include <stdio.h>
int main()
{
    int x=-32769;
    printf("%d", x);
    return 0;
}
```

(Assume integer is of 2 bytes)

The output is: _____.

[MCQ]

7. Consider the following two statements:

P: C standard specifies fixed number of bytes for every data type.

Q: The size order for int, short and long data type is short<int<long

Which of the following statements is/are CORRECT?

- (a) Only P
- (b) Only Q
- (c) Neither P nor Q
- (d) Both P and Q

[MSQ]

8. Which of the following is/are valid declaration of a signed short integer?

- (a) short int a;
- (b) short a;
- (c) signed short a;
- (d) signed short int a;



Answer Key

- | | |
|---|--|
| 1. (c)
2. (-32767)
3. (b)
4. (d) | 5. (-125)
6. (32767)
7. (b)
8. (a, b, c, d) |
|---|--|



Hints and Solutions

1. (c)

Both P and Q are valid declarations.

2. (-32767)

32769 is 2 steps ahead of 32767. After 32767, 2 steps are counted from -327688(including -32768) as
-32768, -32767

Printed value = -32767.

3. (b)

141 is 14 steps ahead of 127. After 127, 14 steps are counted from -128(including -128) as
-128, -127, -126, -125, -124, -123, -122, -121, -120,
-119, -118, -117, -116, -115.

Printed value = -115.

4. (d)

Unsigned value for -134= $256 - 134 = 122$.
Hence, 'z' is printed.

5. (-125)

$$ch = 125 + 6 = 131$$

131 is 4 steps ahead of 127. After 127, 4 steps are counted from -128(including -128) as

-128, -127, -126, -125

Output = -125.

6. (32767)

Printed value = $65536 - 32769 = 32767$.

7. (b)

P: INCORRECT. C standard does not specify fixed number of bytes for any data type. The number of bytes for any data type depends on compiler.

Q: CORRECT. The size order for int, short and long data type is short<int<long.

8. (a, b, c, d)

All are valid declarations.



Any issue with DPP, please report by clicking here:- <https://forms.gle/t2SzQVvQcs638c4r5>

For more questions, kindly visit the library section: Link for web: <https://smart.link/sdfez8ejd80if>



PW Mobile APP: <https://smart.link/7wwosivoicgd4>

C Programming

Data Types & Operators

DPP-02

[MCQ]

```
1. include <stdio.h>
int main(void){
    int a;
    a = 2 * 6/5 + 3.0/2 + 1;
    printf("%d", a);
    return 0;
}
```

The value of a is ____

(a) 4.9 (b) 4.0
 (c) 4.5 (d) 4

[MCQ]

```
2. #include <stdio.h>
int main(void){
    int a;
    a = 16.0 / 4 * 5 % 3;
    printf("%d", a);
    return 0;
}
```

The value of a printed is ____

(a) Compiler error (b) 8.0
 (c) 2 (d) 8

[NAT]

3. Consider the following program.

```
#include<stdio.h>
void main()
{
    int a;
    a=32>24>13>10>8>-1>0;
    printf("%d",a);
}
```

The output is ____.

[NAT]

4. #include<stdio.h>

```
void main()
{
    int a;
```

```
a=25>15>0!=12<45>42!
= 65;
printf("%d",a);
}
```

The output is ____.

[MCQ]

5. Consider the following program:

```
#include<stdio.h>
void main()
{
    int a=0, b=1;
    a=(a==5)&&(b==0);
    printf("%d", a);
    printf("%d", b);
}
```

The output is:

- (a) 50 (b) 00
 (c) 10 (d) Compiler error

[MCQ]

6. Consider the following statements:

P: The precedence of the modulus operator is higher than multiplication or division operator.

Q: The result of the modulus operator contains the sign of the second operand.

Which of the following statements is/are INCORRECT?

- (a) Only P
 (b) Only Q
 (c) Both P and Q
 (d) Neither P nor Q

[MCQ]

7. Consider the following program:

```
#include<stdio.h>
void main()
{
    int a=2022;
```

```
printf("%d%d%d", a!=2024, a==2023, a==2021);  
}
```

The output is-

- (a) 020220
- (b) 020231
- (c) 002021
- (d) 120230

```
void main()  
{  
    int x=-2023;  
    printf("%d", ~x=x+5));  
}
```

The output is _____.

[NAT]

8. Consider the following program:

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



Answer Key

- 1. (d)
- 2. (a)
- 3. (1)
- 4. (1)

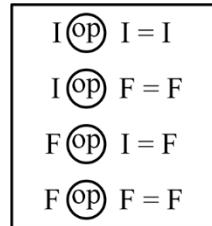
- 5. (b)
- 6. (c)
- 7. (d)
- 8. (2017)



Hints and Solutions

1. (d)

$$\begin{aligned} a &= 12/5 + 3.0/2 + 1 \\ &= 2 + 1.5 + 1 \\ &= 4.5 \\ a \text{ is an integer, so } a &= 4 \end{aligned}$$



2. (a)

$$\begin{aligned} a &= 16.0/4 * 5 \% 1 \\ &= 4.0 * 5 \% 3 \\ &= 20.0 \% 3 \\ &= \boxed{\text{ERROR}} \end{aligned}$$

Modulus operator works only with integers.

3. (1)

$$a = 32 > 24 > 13 > 10 > 8 > -1 > 0$$

$$\begin{array}{c} 1 > 13 \Rightarrow 0 > 10 \\ \quad 0 > 8 \\ \quad \quad 0 > -1 \\ \quad \quad \quad 1 > 0 \\ \quad \quad \quad \quad 1 \end{array}$$

$$a = 1$$

4. (1)

$$a = 25 > 15 > 0! = 12 < 45 > 42! = 65$$

$$\begin{array}{c} 1 > 0 \\ 1! = 12 \\ 1 < 45 \end{array}$$

$$1 > 42 \Rightarrow 0! = 65$$

$$1$$

$$a=1;$$

Output: 1

5. (b)

`int a=0, b=1;`

`a=(a=5)&&(b=0);`

// Assignment operator assigns and returns the assigned value. So, a=5&&0=0, b=0

`printf("%d", a);`//0 is printed

`printf("%d", b);`//0 is printed.

6. (c)

P: INCORRECT. The precedence of the modulus operator is same as multiplication or division operator.

Q: INCORRECT. The result of the modulus operator contains the sign of the first operand.

7. (d)

`a=2022.` So, `a!=2024` evaluates to 1.

`a=2023.` Assignment operator assigns the value and returns the assigned value.

`a==2021` is equivalent to `2023==2021`. So, it evaluates to 0.

Output: 120230

8. (2017)

$x=x+5 \rightarrow x=-2023+5=-2018$

$\sim(x) \rightarrow \sim(-2018) = -(-2018+1) = 2017.$

Output: 2017.



Any issue with DPP, please report by clicking here:- <https://forms.gle/t2SzQVvQcs638c4r5>

For more questions, kindly visit the library section: Link for web: <https://smart.link/sdfez8ejd80if>



PW Mobile APP: <https://smart.link/7wwosivoicgd4>

C Programming

Functions & Storage Classes

DPP-01

[NAT]

1. Consider the following program:

```
#include<stdio.h>
int f2(int a){
    int b=0;
    b=b+5;
    return a*b;
}
int f1(int a){
    int b;
    b=f2(a);
    return a*b;
}
int main(){
    int i, a=5, b=4;
    for(i=0;i<2;i++){
        b-=f1(a)-f2(a);
        printf("%d\t", b);
    }
    return 0;
}
```

The sum of the printed values is _____

[MCQ]

2. Consider the following program:

```
#include<stdio.h>
void print(int n){
    for(n++;n++;n++)
        printf("GATE Wallah");
}
int main(){
    void print();
    void print();
```

```
print(-9);
return 0;
```

}

Which of the following is correct?

- (a) Compilation error
- (b) “GATE Wallah” will be printed infinite number of times.
- (c) “GATE Wallah” will be printed 5 times.
- (d) “GATE Wallah” will be printed 4 times.

[MCQ]

3. Consider the following program.

```
#include<stdio.h>
void f(int n){
    switch(n<<1+n){
        default: printf("Sresth");
        case 4: printf("Parakram");
        case 3: printf("2024");
        break;
        case 2: printf("2025");
    }
}
int main(){
    f(1);
    return 0;
}
```

The output is-

- (a) Parakram2024
- (b) SresthParakram2024
- (c) Parakram
- (d) Sresth2025

[MCQ]

4. Consider the following program:

```
#include<stdio.h>
void f(){
int x;
x=10<5?printf("%d",
printf("GATE")):printf("")?printf("2024"):printf("%d
",printf("Wallah Parakram"));
}
int main(){
f();
return 0;
}
```

The output is-

- (a) GATE2024
- (b) Wallah Parakram15
- (c) GATEWallah Parakram4
- (d) GATE4

[NAT]

5. Consider the following program:

```
#include<stdio.h>
int f(int b, int a){
int x;
x=a<<b;
b=x*a--;
return a+b-x;
}
```

```
int main(){
```

```
printf("%d", f(1,2));
return 0;
}
```

The value printed is _____.

[MCQ]

6. Consider the following program:

```
#include <stdio.h>
int r(int num){
return --num;
}
int main(){
int n=4;
for (r(n);r(n++);r(--n))
printf("%d\t",r(--n));
return 0;
}
```

The output is-

- (a) 1 2 3
- (b) 1 2 3 4
- (c) 3 2 1
- (d) 4 3 2 1

Answer Key

- 1. (-292)
- 2. (d)
- 3. (a)

- 4. (b)
- 5. (5)
- 6. (c)



Hints and Solutions

1. (-292)

For i=0:

f1(5):

Line 1: int b;

Line 2: b=f2(5);//b=25

Line 3: return 5*25; //return 125 to main().

f2(5):

Line 1: int b=0;

Line 2: b=b+5;//b=5

Line 3: return 5*5;//return 25 to f1. Go to Line 3 of f1(5)

f2(5):

Line 1: int b=0;

Line 2: b=b+5;//b=5

Line 3: return 5*5;//return 25 to main().

b in main() is updated to: $b = b - f1(a) + f2(a) = 4 - 125 + 25 = -96$.

For i=1:

f1(5):

Line 1: int b;

Line 2: b=f2(5);//b=25

Line 3: return 5*25; //return 125 to main().

f2(5):

Line 1: int b=0;

Line 2: b=b+5;//b=5

Line 3: return 5*5;//return 25 to f1. Go to Line 3 of f1(5)

f2(5):

Line 1: int b=0;

Line 2: b=b+5;//b=5

Line 3: return 5*5;//return 25 to main().

b in main() is updated to: $b = b - f1(a) + f2(a) = -96 - 125 + 25 = -196$.

Output is: -96 -196

Sum= -292

2. (d)

```
int main(){
```

```
void print();//No compilation error
```

```
void print();//No compilation error
```

```
print(-9); //print(-9) is called.
```

```
return 0;
```

```
}
```

```
print(-9){//n=9
```

```
for(n++; n++ ;n++)
```

-9	-8	->printf() is executed	-7
----	----	------------------------	----

-6	->printf() is executed	-5
----	------------------------	----

-4	->printf() is executed	-3
----	------------------------	----

-2	->printf() is executed	-1
----	------------------------	----

0	-> Loop terminates	
---	--------------------	--

```
}
```

“GATE Wallah” will be printed four times.

3. (a)

f(1):

n=1;

```
switch(n<<1+n){
```

```
//switch(1<<2) i.e switch(4)
```

```
default: printf("Sresth");
```

```
case 4: printf("Parakram");
```

```
//case 4 is executed.
```

```
//since no break is there case 3 will also be
executed.
```

```
case 3: printf("2024");
```

```
break;
```

```
case 2: printf("2025");
```

```
}
```

Output: Parakram2024

4. (b)

f():

```
x=10<5?printf("%d", printf("GATE"));
printf("")?printf("2024");
printf("%d", printf("Wallah Parakram"));
```

10<5 is FALSE. So, printf("") is evaluated. It prints nothing and hence returns 0.

0 means FALSE. So, printf("%d",printf("Wallah Parakram")) is evaluated.

Output: Wallah Parakram15

5. (5)

f(1,2):

b=1, a=2;

x=a<<b; //x=2<<1= 4

b=x*a--;//b=4*2=8. After this, a is decremented to 1.
return a+b-x; // return 1+8-4 i.e. return 5.

main():

printf("%d", f(1,2));//5 is printed.

Output: 5

6. (c)

r(4)=3. //Initialization

r(n++)or r(4)=3->TRUE// Condition check
n is incremented to 5.

```
printf("%d\t",r(--n));// printf("%d\t",r(4))
//3 is printed.
```

r(--n) or r(3) is called.

r(n++)or r(3)=2->TRUE// Condition check
n is incremented to 4.

```
printf("%d\t",r(--n));// printf("%d\t",r(3))
//2 is printed.
```

r(--n) or r(2) is called.

r(n++)or r(2)=1->TRUE// Condition check
n is incremented to 3.

```
printf("%d\t",r(--n));// printf("%d\t",r(2))
//1 is printed.
```

r(--n) or r(0) is called.

r(n++)or r(1)=0->FALSE//Loop terminates.

Output: 3 2 1



Any issue with DPP, please report by clicking here:- <https://forms.gle/t2SzQVvQcs638c4r5>

For more questions, kindly visit the library section: Link for web: <https://smart.link/sdfez8ejd80if>



PW Mobile APP: <https://smart.link/7wwosivoicgd4>

Programming in C

Functions & Storage Classes

DPP-02

[MCQ]

1. Consider the following program:

```
#include <stdio.h>
int main(){
    int a=2, i;
    static int b=a++;
    for(i=0;i<a+b;i++)
        printf("GATE Wallah");
    return 0;
}
```

Which of the following is/are CORRECT?

- (a) GATE Wallah is printed 3 times.
- (b) GATE Wallah is printed 4 times.
- (c) GATE Wallah is printed 5 times.
- (d) Compilation Error.

[MCQ]

2. #include<stdio.h>

```
void main(){
    extern int a;
    extern int a;
    extern int a;
    printf("%d", a);
}
```

int a = 15;

The output is-

- (a) Garbage value
- (b) Compilation error
- (c) 15
- (d) No output

[NAT]

3. #include<stdio.h>

```
int func(int a, int b){
    static int p=9, q=21;
    if(a>b){
```

```
a=a-p++;
b=b+q--;
```

```
return a+b;
```

```
}else{
```

```
return p-q;
```

```
}
```

```
int main(){
```

```
int i=2, j=-2;
```

```
for(;j<3;j++)
```

```
printf("%d\t",func(i,j));
```

```
return 0;
```

```
}
```

The sum of the values printed is _____.

[MCQ]

4. Consider the following program:

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

```
void f(){
```

```
static int a=3;
```

```
int b=5;
```

```
a=b++;
```

```
printf("%d\t%d\n",a,b);
```

```
}
```

```
int main()
```

```
{
```

```
static int a=2;
```

```
int b=1;
```

```
f();
```

```
a+=3;
```

```
f();
```

```
printf("%d\t%d", a,b);
```

```
return 0;
```

```
}
```

The output is-

(a)

```
-2 6
-7 6
5 1
```

(b)

```
-2 6
-7 7
5 1
```

(c)

```
-2 5
-7 6
5 1
```

(d) None

[NAT]

5. #include <stdio.h>

```
int i=1;
int f(){
    static int i=2;
    return i++;
}
int main()
{
    extern int i;
    char a='B';
    printf("%d",a+f()+f()+i);
    return 0;
}
```

The output is _____

[MCQ]

6. Consider the following program:

```
#include <stdio.h>
static int j;
```

static int j=3;

```
int f(){
    auto int i=2;
    return i++++-j;
}
```

int main()

```
{
    char a='B';
    printf("%d",a+f()+f());
    return 0;
}
```

The output is-

(a) 68

(b) 72

(c) 73

(d) Compilation error

[NAT]

7. Consider the following program:

```
#include <stdio.h>
int f(){
    static int i=5;
    return i--;
}
int main()
{
    for(f();f();printf("GATE Wallah"))
        printf("Pankaj Sharma");
    return 0;
}
```

The number of times printf() is executed is _____.

[MSQ]

8. Which of the following statement(s) is/are CORRECT?

(a) A static variable has internal linkage.

(b) Static variables are stored in the data segment.

(c) Auto variables are stored in the heap segment.

(d) Register variables behave as auto variables by default.

Answer Key

- | | |
|---------------------------------------|---|
| 1. (d)
2. (c)
3. (38)
4. (a) | 5. (72)
6. (c)
7. (8)
8. (a, b, d) |
|---------------------------------------|---|



Hints and Solutions

1. (d)

static int b=a++; //This is not allowed at the time of declaration of static variable. Assigned value must be a constant.
Hence, compilation error exists.

2. (c)

'extern int a' can be written multiple times.
extern shares the space of global variables.
∴ Output: 15

3. (38)

func(i, j) will be called 5 times for j values IN {-2, -1, 0, 1, 2}

Data segment	
p: 9 10 11 12 13	q: 21 20 19 18 17

func(2, -2):

a=2, b=-2

```
if(a>=b){ (2>-2)->TRUE
    a=a-p++; // a=2-9=-7;p is then incremented to 10
    b=b+q--; //b=-2+21=19;q is then decremented to
    20
    return a+b; //return (-7+19) i.e 12; So, 12 will be
    printed
} else return p-q;
```

Similarly, func(2, -1) returns 11; func(2, 0) returns 10; func(2, 1) returns 9;

func(2, 3) returns p-q i.e 13-17 i.e -4

Output: 12 11 10 9 -4

Sum: 38

4. (a)

f():

```
static int a=3;
int b=5;
a=b++; //a=3-5=-2; b is incremented to 6.
printf("%d\t%d\n",a,b); //2 6 is printed.
```

f():

```
static int a=3; // a contains -2.
int b=5;
a=b++; //a=-2-5=-7; b is incremented to 6.
printf("%d\t%d\n",a,b); // -7 6 is printed.
```

main():

```
static int a=2;
int b=1;
a+=3; //a=5
printf("%d\t%d\n",a,b); //5 1 is printed.
```

Output:

```
-2 6
-7 6
5 1
```

5. (72)

f():

```
static int i=2;
return i++; // return 2; i is incremented to 3.
```

f():

```
static int i=2; //i contains 3.
return i++; // return 3; i is incremented to 4.
```

main():

```
extern int i; //extern variable shares the space of global
variables.
char a='B';
printf("%d",a+f()+f()+i);//(66+2+3+1) i.e. 72 is
printed.
return 0;
```

6. (c)

static int j;

static int j=3;

Multiple declarations of global static variables are allowed. Hence, no compilation error.

f():

```
auto int i=2; // i is an auto or local variable.
return i+++--j; // return 2+2 i.e. 4. Auto i is incremented
to 3, static j contains 2.
```

f():

auto int i=2; // i is an auto or local variable. It will be re-initialized to 2

return i+++-j; // return 2+1 i.e. 3. Auto i is incremented to 3, static j contains 1.

main():

char a='B';

printf("%d",a+f()+f()); // (66+4+3) i.e 73 is printed.

Output: 73

7. (8)

Initialization: f() returns 5; Static i is decremented to 4.

Condition: f() returns 4-> TRUE. Static i is decremented to 3.

Body: "Pankaj Sharma" is printed.

"GATE Wallah" is printed.

Condition: f() returns 3-> TRUE. Static i is decremented to 2.

Body: "Pankaj Sharma" is printed.

"GATE Wallah" is printed.

Condition: f() returns 2-> TRUE. Static i is decremented to 1.

Body: "Pankaj Sharma" is printed.

"GATE Wallah" is printed.

Condition: f() returns 1-> TRUE. Static i is decremented to 0.

Body: "Pankaj Sharma" is printed.

"GATE Wallah" is printed.

Condition: f() returns 0-> FALSE. Execution stops.

Hence, printf() is executed 8 times.

8. (a, b, d)

(a) CORRECT. A static variable has internal linkage.

(b) CORRECT. Static variables are stored in the data segment.

(c) INCORRECT. Auto variables are stored in the stack segment.

(d) CORRECT. Register variables behave as auto variables by default.



Any issue with DPP, please report by clicking here:- <https://forms.gle/t2SzQVvQcs638c4r5>

For more questions, kindly visit the library section: Link for web: <https://smart.link/sdfez8ejd80if>



PW Mobile APP: <https://smart.link/7wwosivoicgd4>

C Programming

Functions & Storage Classes

DPP-03

[NAT]

1. Consider the following function:

```
int f(int n)
{
    if(n<2) return n;
    return f(n/3)+3;
}
```

The value returned by f(27) is _____

[MCQ]

2. Consider the following program:

```
#include<stdio.h>
static int i;
extern int i=3;
void f(int n)
{
    if(n)
    {
        printf("%d\t", i++);
        f(n-1);
    }
}
int main()
{
    f(4);
    return 0;
}
```

The output is:

(a) 3 4 5 6 7
 (b) 3 4 5 6
 (c) 0 1 2 3
 (d) Compilation error.

[NAT]

3. Consider the following function:

```
void f(int n){
static int i=3;
    if(n%2){
        n+=i--;
        printf("%d\t", n);
        f(n-1);
    }
}
```

```
}
```

```
n++;
```

```
printf("%d\t", n);
```

```
}
```

The sum of the values printed when f(5) is called is _____

[NAT]

4. Consider the following function:

```
int func(int a, int b)
{
    if (a<b)
        return a+b;
    else
        return 1+func(a-1, b+1);
}
```

The value returned by func(4, 2) is _____.

[MCQ]

5. Consider the following function:

```
void arc(int n){
    if (n<=2) return;
    else{
        arc(n-2);
        printf("%d\t", n-1);
        arc(n-3);
        printf("%d\t", n-2);
    }
}
```

The output printed when arc(7) is called-

(a) 6 2 1 4 3 3 2 5

(b) 2 3 4 1 2 3 6 5

(c) 2 1 4 3 6 3 2 5

(d) 5 2 3 6 3 4 2 1

[MCQ]

```

6. #include<stdio.h>
void print(int n)
{
    printf("GATE Wallah\n");
    if(n++==0) return;
    print(n++);
}
int main()
{
    void print();
    void print();
    print(-4);
    return 0;
}

```

Which of the following is correct?

- (a) Compilation error
- (b) “GATE Wallah” will be printed infinite number of times.
- (c) “GATE Wallah” will be printed 4 times.
- (d) “GATE Wallah” will be printed 5 times.

[MCQ]

7. Consider the following program:

```

void arc()
{
    int a;
    static int b;
    a=b;
    printf("%d\t%d\t", a++, b++);
    if(a<=3)
        arc();
}

```

When arc() is called from main, the output is:

- (a) 00112233
- (b) 11223344
- (c) 01122334
- (d) 12233445

[NAT]

8. Consider the following function:

```

int arc(int i, int j)
{
    if(i<2) return j+2;
    else if(j<2) return arc(i-1, 1);
    else return arc(i-1, arc(i, j-2));
}

```

The value returned by arc(2, 6) is _____

Answer Key

- 1. (10)
- 2. (b)
- 3. (45)
- 4. (8)

- 5. (c)
- 6. (d)
- 7. (a)
- 8. (9)



Hints and Solutions

1. (10)

f(27):

if($n < 2$) return n ; // $27 < 2 \rightarrow \text{FALSE}$

return $f(n/3) + 3$; // $f(9)$ is called;

// $f(27)$ returns $(7+3)$ i.e 10

$f(9)$:

if($n < 2$) return n ; // $9 < 2 \rightarrow \text{FALSE}$

return $f(n/3) + 3$; // $f(3)$ is called;

// $f(9)$ returns $(4+3)$ i.e 7 to $f(27)$

$f(3)$:

if($n < 2$) return n ; // $3 < 2 \rightarrow \text{FALSE}$

return $f(n/3) + 3$; // $f(1)$ is called; $f(1)$ returns 1.

// $f(3)$ returns $(1+3)$ i.e 4 to $f(9)$

2. (b)

extern int $i=3$; //No compilation error. Assigns 3 to global static variable i .

$f(4)$:

if(4){//true

printf("%d\t", $i++$); //3 is printed. Global static i is incremented to 4.

$f(n-1)$;// $f(3)$ is called

}

$f(3)$:

if(3){//true

printf("%d\t", $i++$); //4 is printed. Global static i is incremented to 5.

$f(n-1)$;// $f(2)$ is called

}

$f(2)$:

if(2){//true

printf("%d\t", $i++$); //5 is printed. Global static i is

incremented to 6.

$f(n-1)$;// $f(1)$ is called

}

$f(1)$:

if(1){//true

printf("%d\t", $i++$); //6 is printed. Global static i is incremented to 7.

$f(n-1)$;// $f(0)$ is called. It does nothing.

}

Output: 3 4 5 6

3. (45)

4. (8)

func($4, 2$):

$4 < 2 \rightarrow \text{FALSE}$

return $1 + \text{func}(3, 3)$;// return $(1+7)$ i.e return 8

func($3, 3$):

$3 < 3 \rightarrow \text{FALSE}$

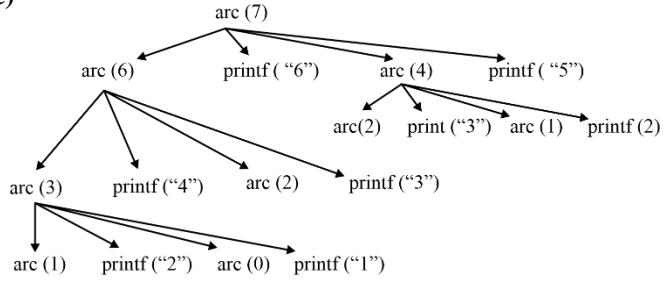
return $1 + \text{func}(2, 4)$;// return $(1+6)$ i.e return 7

func($2, 4$):

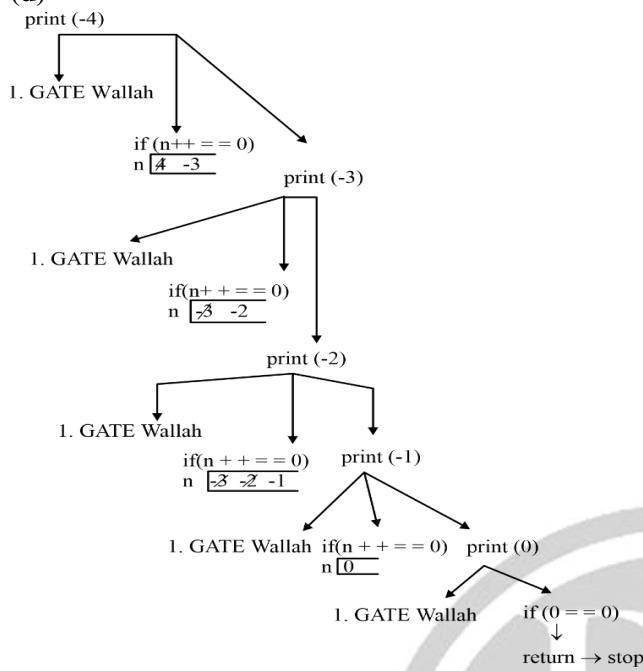
$2 < 4 \rightarrow \text{TRUE}$

return $(2+4)$;// return 6

5. (c)



6. (d)



“GATE Wallah” is printed 5 times.

7. (a)

arc():

```

int a;
static int b; // static variable b is by default initialized to 0
a=b;// a contains 0
printf("%d\t%d\t", a++, b++); // 0 0 is printed. a is incremented to 1, b is incremented to 1.
if(a<=3) //1<=3-> True
arc(); // arc() is called.
  
```

arc();

```

int a;
static int b; // static variable b contains 1.
a=b;// a contains 1
printf("%d\t%d\t", a++, b++); // 1 1 is printed. a is incremented to 2, b is incremented to 2.
if(a<=3) //2<=3-> True
arc(); // arc() is called.
  
```

arc():

```

int a;
  
```

```

static int b; // static variable b contains 2.
  
```

```

a=b;// a contains 2
  
```

```

printf("%d\t%d\t", a++, b++); // 2 2 is printed. a is incremented to 3, b is incremented to 3.
  
```

```

if(a<=3) //3<=3-> True
  
```

```

arc(); // arc() is called.
  
```

arc():

```

int a;
  
```

```

static int b; // static variable b contains 3.
  
```

```

a=b;// a contains 3.
  
```

```

printf("%d\t%d\t", a++, b++); // 3 3 is printed. a is incremented to 4, b is incremented to 4.
  
```

```

if(a<=3) //4<=3-> False; execution stops.
  
```

8. (9)

arc(2, 6):

1: return arc(1, arc(2, 4)); //arc(2, 4) returns 7

7: return arc(1, 7); //return (7+2) i.e 9

arc(2,4):

2: return arc(1, arc(2, 2)); //arc(2, 2) returns 5 to arc(2, 4)

6: return arc(1, 4); //arc(1, 4) returns 7.

arc(2, 2):

3: return arc(1, arc(2, 0));
//arc(2, 0) returns 3 to arc(2, 2)

5: return arc(1, 2); //arc(1,2) returns 5

arc(2,0):

4: return arc(1, 1); //arc(1, 1) returns (1+2) i.e 3 to arc(2, 0)



Any issue with DPP, please report by clicking here:- <https://forms.gle/t2SzQVvQcs638c4r5>

For more questions, kindly visit the library section: Link for web: <https://smart.link/sdfez8ejd80if>



PW Mobile APP: <https://smart.link/7wwosivoicgd4>

C Programming

Functions & Storage Classes

DPP-04

[NAT]

1. Consider the following function:

```
int func(int a)
{
    static int b=1;
    b=b+a;
    if((b%a)%2!=0) return a+func(b+a);
    return b-a;
}
```

The value returned by func(5) is _____.

[MCQ]

2. Consider the following function:

```
void func(int n)
{
    if(n>0){
        func(n-1);
        printf("%d\t", n);
    }
    printf("%d\t", n-1);
}
```

The output printed by func(2) is-

- (a) -1 1 0 2 1 (b) -1 1 0 2 1 2
 (c) -1 1 0 2 -2 1 (d) -1 1 0 2 1

[NAT]

3. Consider the following function:

```
int func(int n)
{
    if(n>0){
        return func(n/2)+func(n/4)+1;
    }
    return n+1;
}
```

The value returned by func(6) is _____

[NAT]

4. Consider the following function:

```
int func(int n)
{
    if(n>0){
        return 3*func(n/4)+1;
    }
    return n;
}
```

The value returned by func(24) is _____.

[NAT]

5. Consider the following function:

```
int func(int n)
{
    static int k=0;
    if(n>0){
        k++;
        return 2*func(n/2)+k;
    }
    return n+k--;
}
```

The value returned by func(8) is _____.

[MCQ]

6. Consider the following function:

```
int func(int n, int i)
{
    if(n==0) return 0;
    else if(n%2){
        return func(n/2, 2*i)+i;
    }else return func(n/2, 2*i)-i;
}
```

The value returned by func(14, 1) is-

- (a) 1 (b) 13
 (c) 15 (d) 0

[MCQ]

7. Consider the following function:

```
int func(int n)
{
    static int i=0;
    if(n/2){
        i--;
        return func(n/2)+i;
    }else return i;
}
```

The value returned by func(7) is-

- (a) -6
- (b) -12
- (c) -18
- (d) -21

[MCQ]

8. Consider the following function:

```
void display()
{
    static int i;
    if(i<=printf("GATE24")){
        i=i+2;
        display();
    }
}
int main()
{
    int i=0;
    for(i=0;i<3;i++)
        display();
    return 0;
}
```

The number of times printf() executed is-

- | | |
|-------|-------|
| (a) 6 | (b) 5 |
| (c) 7 | (d) 9 |

Answer Key

- 1. (11)
- 2. (d)
- 3. (9)
- 4. (13)

- 5. (109)
- 6. (b)
- 7. (a)
- 8. (c)



Hints and Solutions

1. (11)

func(5):

```
static int b=1; //static b is initialized to 1.
b=b+a; //b=1+5=6
if((b%a)%2!=0) //(6%5)%2!=0 is TRUE
Line 1: return a+func(b+a); // return 5+func(6+5);
func(11) is called. Returns 5+6 i.e 11.
```

Line 2: return b-a;

func(11):

```
static int b=1; //static b is initialized to 1.
b=b+a; //b=6+11=17
if((b%a)%2!=0) //(17%11)%2!=0 is FALSE
Line 1: return a+func(b+a);
Line 2: return b-a; // return (17-11) i.e 6 to Line 1
of func(5);
```

2. (d)

func(2):

2>0 True

func(1) is called.

printf("%d\t", n); // 2 is printed.

printf("%d\t", n-1); //1 is printed.

func(1):

1>0 True

func(0) is called.

printf("%d\t", n); // 1 is printed.

printf("%d\t", n-1); //0 is printed.

func(0):

0>0 is FALSE

printf("%d\t", n-1); // -1 is printed.

Output: -1 1 0 2 1

3. (9)

func(6):

```
if(n>0){ //6>0 is TRUE
1. return func(n/2)+func(n/4)+1;//return
func(3)+func(1)+1; return 5+3+1=9;
}
```

func(3):

```
if(n>0){ //3>0 is TRUE
1. return func(n/2)+func(n/4)+1;//return
func(1)+func(0)+1; return 3+1+1=5;
}
```

func(1):

```
if(n>0){ //3>0 is TRUE
1. return func(n/2)+func(n/4)+1;//return
func(0)+func(0)+1;//return 3;
}
```

func(0):

return 1;

4. (13)

func(24):

```
if(n>0){ //24>0 is TRUE
return 3*func(n/4)+1; //func(6) is called.
Returns 3*4+1; Returns 13.
}
```

func(6):

```
if(n>0){ //6>0 is TRUE
return 3*func(n/4)+1; //func(1) is called.
Returns 3*1+1; returns 4;
}
```

func(1):

```
if(n>0){ //1>0 is TRUE
return 3*func(n/4)+1; //func(0) is called.
Returns 1;
}
```

func(0):

return 0;

5. (109)

```

func(8):
    static int k=0;
    if(n>0){//8>0 is TRUE
        k++; //static k is incremented to 1.
        return 2*func(n/2)+k;//func(4) is called.
        Returns (2*53+3) i.e 109
    }

func(4):
    static int k=0;
    if(n>0){//4>0 is TRUE
        k++; //static k is incremented to 2.
        return 2*func(n/2)+k;//func(2) is called.
        Returns (2*25+3) i.e 53
    }

func(2):
    static int k=0;
    if(n>0){//2>0 is TRUE
        k++; //static k is incremented to 3.
        return 2*func(n/2)+k;//func(1) is called.
        Returns (2*11+3) i.e 25
    }

func(1):
    static int k=0;
    if(n>0){//1>0 is TRUE
        k++; //static k is incremented to 4.
        return 2*func(n/2)+k;//func(0) is called.
        Returns(2*4+3) i.e 11
    }
f(0) returns (0+4). Static k is decremented to 3.

```

6. (b)

```

func(14, 1):
    14%2 is 0, so else part is executed.
    return func(7, 2)-1;// //Returns 14-1 i.e 13.

func(7,2):
    7%2 is 1, so else if part is executed.
    return func(3, 4)+2; //Returns 12+2 i.e 14 to
    func(14, 1)

func(3,4):
    3%2 is 1, so else if part is executed.
    return func(1, 8)+4; //Returns 8+4 i.e 12 to func(7, 2)

```

func(1,8):

1%2 is 1, so else if part is executed.
 return func(0, 16)+8; //fun(0,16) returns 0
 //Returns 8 to func(3, 4)

7. (a)

```

func(7):
    static int i=0;
    if(n/2){//7/2= 3 is TRUE
        i--;/static i is decremented to -1
        return func(n/2)+i; //func(3) is called. func(7)
        returns -4-2 i.e -6
    }else return i;

func(3):
    static int i=0;
    if(n/2){//3/2= 1 is TRUE
        i--;/static i is decremented to -2
        return func(n/2)+i; //func(1) is called. func(1)
        returns -2. func(3) returns -2-2 i.e -4
    }else return i;

```

8. (c)

For i=0 in main():
 display():

```

        static int i;/i=0
        if(i<=printf("GATE24")){//i<=6; printf() executed
            i=i+2;/i=2
            display();
        }
    
```

display():

```

        static int i;
        if(i<=printf("GATE24")){//2<=6; printf() executed
            i=i+2;/i=4
            display();
        }
    
```

display():

```

        static int i;
        if(i<=printf("GATE24")){//4<=6; printf() executed
            i=i+2;/i=6
            display();
        }
    
```

```

display():
static int i;
if(i<=printf("GATE24")){//6<=6; printf() executed
    i=i+2;//i=8
    display();
}
display():
static int i;
if(i<=printf("GATE24")){//8<=6  is FALSE  but
    printf() executed
    i=i+2;
    display();
}

```

For i=1 in main:

```

display():
static int i;
if(i<=printf("GATE24")){//8<=6  is FALSE  but
    printf() executed
    i=i+2;
    display();
}

```

For i=2 in main():

```

display():
static int i;
if(i<=printf("GATE24")){//8<=6  is FALSE  but
    printf() executed
    i=i+2;
    display();
}

```

Total number of times printf() executed is 7.



Any issue with DPP, please report by clicking here:- <https://forms.gle/t2SzQVvQcs638c4r5>

For more questions, kindly visit the library section: Link for web: <https://smart.link/sdfez8ejd80if>



PW Mobile APP: <https://smart.link/7wwosivoicgd4>

Programming in C

Strings

DPP-01

[NAT]

1. Consider the following codes P and Q as:

```
P : char* p ="GATEWallah";
    p[5]= 'A';
    printf("%s",p);
Q: char* p ="GATEWallah";
    char* q = p;
    q[5]= 'A';
    printf("%s",q);
```

The number of INCORRECT codes is/are _____.

[MCQ]

2. P : char s1[]="GATE";
 char s2[]="GATE";
 if(s1==s2) printf("YES");
 else
 printf("NO");
Q : char s1[]="GATE";
 char s2[]="GateWallah";
 if(*s1==*s2) printf("YES");
 else
 printf("NO");

The outputs are-

- (a) P = YES Q = YES
- (b) P = YES Q = NO
- (c) P = NO Q = YES
- (d) P = NO Q = NO

[MCQ]

3. P : char s[20];
 printf("Enter your GATE stream with year: \n");
 scanf("%s",s);
 printf("%s",s);
Q : char s[20];
 printf("Enter your GATE stream with year: \n");
 gets(s);
 printf("%s",s);

If the input string is “CS 2023”, the outputs are-

- (a) P=CS 2023 Q = CS 2023
- (b) P=CS Q = CS

- (c) P=CS 2023 Q = CS
- (d) P = CS Q = CS 2023

[MCQ]

```
4. #include<stdio.h>
#include<string.h>
int main()
{
    char s[20]="GATEWallah";
    printf("%s",s+4);
    s[4]=0;
    printf("%s",s);
    return 0;
}
```

The output is-

- (a) WallahGATE
- (b) EWallahGAT
- (c) WallahGATE0allah
- (d) EWallahGAT0allah

[MCQ]

```
5. #include<stdio.h>
#include<string.h>
int main()
{
    char s[20]="GATEWallah2023";
    s[10]='0';
    printf("%s",s+s[3]-s[1]);
    return 0;
}
```

The output printed is-

- (a) Wallah0 (b) Wallah2023
- (c) Wallah0023 (d) Wallah

[MCQ]

```

6. #include<stdio.h>
#include<string.h>
void f(char *p)
{
    static int q=2;
    q=q+3;
    p[q]+=2;
}
int main()
{
    char s[20]="GATEWallahbesthai";
    int i=0;
    for(i=0;i<3;i++){
        f(s);
    }
    printf("%s",s);
    return 0;
}

```

The output string printed is-

- (a) GATEWcllchbgsthai
- (b) GATEWcllbhbgssthai
- (c) GATEWcllchbesthai
- (d) GATEWcllchbesthai

[MCQ]

```

7. #include<stdio.h>
#include<string.h>
void f(char *p){
    if(*p!=0){
        printf("%c", *p);
        f(p+1);
    }
    printf("%c", *p);
}
int main()
{
    char s[5]="GATE";
    f(s);
    return 0;
}

```

The output is-

- (a) GATEGATE
- (b) ETAGGATE
- (c) ETAGETAG
- (d) GATEETAG

[NAT]

```

8. #include<stdio.h>
#include<string.h>
int main()
{
    int a=1;
    char b[]="GATE2024";
    char c[]="GATE2024";
    int d=strcmp(b,c);
    if(d==0)
        a=printf("GATEWallah");
    printf("%d",a);
    return 0;
}

```

The value of a is_____.

Answer Key

- 1. (2)
- 2. (c)
- 3. (d)
- 4. (a)

- 5. (c)
- 6. (a)
- 7. (d)
- 8. (10)



Hints and solutions

1. (2)

`char*p = "GATEWallah";`

Memory is allocated to "GATEWallah" in static/read only memory. So, its content cannot be updated
 $p[5] = 'A'$

It is not allowed as 'p' is the only entry point to the string constant.

\therefore Both P and Q are not valid.

2. (c)

P: if ($s1 == s2$) // It is comparing the base addresses of two different Strings.

\rightarrow false

\therefore else part will be executed

\downarrow

No is printed

Q:

$s1:$

G A T E
100

$s2:$

G A T E W a l l a h
200

$\text{if } (*s1 == *s2) \Rightarrow \text{if } (*100 == *200)$

$* \rightarrow$ value at

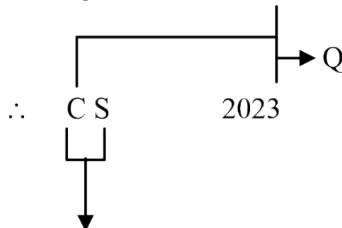
\downarrow

$G == G$

\rightarrow TRUE

3. (d)

`scanf()` halts reading as soon as it encounters whitespace. `gets()` ignores the whitespace and stops reading when new-line is found.



\therefore Output of P: CS

Output of Q: CS 2023

4. (a)

100 101 102 103 104 105 106 107 108 109 110

S: [G A T E W a 1 1 a h \0]
\0

`printf("%s", s + 4); // Wallah`

\downarrow

104

$s[4] = 0; //*(100+4) = 0$ where 0 is the ASCII of NULL character.

`print("%s", s); // It prints the string till it encounters first NULL;`

\Rightarrow Output is: WallahGATE

5. (c)

100 101 102 103 104 105 106 107 108 109 110 111 112 113 114

S: [G A T E W a 1 1 a h Z 0 2 3 \0]
0

$s[10] = '0';$ // Here '0' is the numeric 0

`printf("%s", s+s[3]-s[1]);`

\downarrow

$100 + 69 - 65 = 104$

\therefore Output is: Wallah0023

6. (a)

S:

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
G A T E W a 1 1 a h b e s t h a i

starting address of S: 100

i [0]
f(100)

i [1]
f(100)

p [100]
q [Z 5]

p [100]
q [Z 8]

$$\begin{aligned} p[5] &+= 2; // p[5] = c & p[8] &+= 2; // p[8] = c \\ &&& \end{aligned}$$

i

f(100)

p

q

$p[11] += 2; // p[11] = g$

Output: G A T E W c 11 c h b g s t h a i

7. (d)

G	A	T	E	\0
1 0 0	1 0 1	1 0 2	1 0 3	1 0 4

f(100)

*100==G!=0→True

(1) printf() executed → G
f(101)

(8) printf() executed → G

f(101)

*101==A!=0→True

(2) printf() executed → A
f(102)

(7) printf() executed → A

f(102)

*102==T!=0→True

(3) printf() executed → T
f(103)

(6) printf() executed → T

f(103)

*103==E!=0→True

(4) printf() executed → E
f(104)→NULL is present

(5) printf() executed → E

∴ Output is: GATEETAG

8. (10)

int a=1;

char b[]="GATE2024";

char c[]="GATE2024";

int d=strcmp(b,c);

//When the strings are equal, strcmp returns 0.

if(d==0)

a=printf("GATEWallah");

//printf() returns the number of characters it printed.

printf("%d",a);//10

return 0;



Any issue with DPP, please report by clicking here:- <https://forms.gle/t2SzQVvQcs638c4r5>

For more questions, kindly visit the library section: Link for web: <https://smart.link/sdfez8ejd80if>



PW Mobile APP: <https://smart.link/7wwosivoicgd4>

Programming in C

Strings

DPP-02

[MCQ]

1. Consider the following program:

```
#include<stdio.h>
#include<string.h>
int main()
{
char a[]="GATE_Wallah";
printf("%s\t", &a[5]);
printf("%s", (a+5));
return 0;
}
```

The output is-

- (a) Runtime Error
- (b) Wallah Wallah
- (c) _Wallah_Wallah
- (d) Compilation Error

[MCQ]

2. Consider the following program:

```
#include<stdio.h>
#include<string.h>
int main()
{
char s[5];
s="GATE";
printf("%s",s);
return 0;
}
```

The output is-

- (a) GATE
- (b) G
- (c) NULL
- (d) Compiler Error

[NAT]

3. Consider the following program:

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

```
int main()
{
char *p="abcd";
char *q="acd";
int a;
a=strcmp(p,q)?strlen(p):strlen(q);
printf("%d", a);
return 0;
}
```

The output is _____.

[MCQ]

4. Consider the following program:

```
#include<stdio.h>
#include<string.h>
int main()
{
char * s[5]={"CS", "MECH", "ECE",
"ELECTRICAL", "CIVIL"};
char ** p[5]={s+2, s+4, s+1, s+3, s};
char ***q=p;
q=q+3;
printf("%s", q[-2][-1]);
q=q-2;
printf("%c", ***q++++);
return 0;
}
```

The output printed is-

- (a) CSF
- (b) ELECTRICALD
- (c) CSD
- (d) CIVILF

[MCQ]

5. Consider the following program:

```
#include<stdio.h>
#include<string.h>
int main()
{
char a[]="GATE2024";
```

```

char b[9];
strcpy(b, a);
printf("%s%d%d", b, strlen(b), sizeof(b));
return 0;
}

```

The output is –

- (a) Compilation Error
- (b) GATE202488
- (c) GATE202489
- (d) GATE202499

[MCQ]

6. Consider the following program:

```

#include<stdio.h>
#include<string.h>
void func(char *ptr)
{
    if((*ptr)!='0'){
        printf("%c", *ptr);
        func(ptr+2);
    }
}
int main()
{
    func("GATEWallah");
    return 0;
}

```

The output is _____

- (a) GTWla
- (b) GTWlh
- (c) GATEWallah
- (d) None of the above

[MCQ]

7. Consider the following program:

```

#include<stdio.h>
#include<string.h>
int main()
{
    char a[]="GATEWallah";
    char *p;
    p=a;
    p+=4;
    *p='\0';
    printf("%s",p);
    return 0;
}

```

}

The output is-

- | | |
|---------------|-----------------|
| (a) No output | (b) Wallah |
| (c) \0allah | (d) GATE\0allah |

[NAT]

8. Consider the following program:

```

#include<stdio.h>
#include<string.h>
int main()
{
    char *a[]={ "GATE", "Parakram", "Wallah",
    "Shreshth" };
    char **b[]={ a+2, a+3, a+1, a };
    char ***c=b;
    printf("%s", *-*++c+3);
    printf("%s", **++c);
    return 0;
}

```

The length of the output string is _____.

Answer Key

- 1. (b)
- 2. (d)
- 3. (4)
- 4. (b)

- 5. (c)
- 6. (a)
- 7. (a)
- 8. (11)



Hints and solutions

1. (b)

%s takes the start address and prints until it reaches NULL character.

```
char a[]="GATE_Wallah";
printf("%s\0", &a[5]); Address passed is (a+5).
"Wallah " is printed.
printf("%s", (a+5)); "Wallah" is printed.
```

Output: Wallah Wallah

2. (d)

s means the base address of the string.
s="GATE"; // A string “GATE” is being assigned to an address. Incompatible assignment.

Output: Compilation Error

3. (4)

strcmp(p,q) compares the two strings by taking the base addresses as parameters. It returns 0 only if they are equal.

The strings “abcd” and “acd” are not equal. So strcmp(*p,*q) returns a non-zero value.

Since, the condition is TRUE, strlen(p) is assigned to a.
a=4.

4. (b)

100	400
C S	E L E C T R I C A L

200	300	500
M E C H	E C E	C I V I L

S	100	200	300	400	500
	600	604	608	612	616

p	608	616	604	612	600
	700	704	708	712	716

q	700	712	704
---	-----	-----	-----

$q = q + 3$

printf ("%s", q [-2] [-1]);

$\Rightarrow *(*(q - 2) - 1)$

$\Rightarrow *(*704 - 1)$

$\Rightarrow *(616 - 1)$

$\Rightarrow *(612)$

$\Rightarrow 400 \Rightarrow \text{ELECTRICAL}$

$q = q - 2; // q = 712 - 2 * 4 = 704$

printf ("%c", ***q++ + 1);

$\Rightarrow ***704 + 1$

$\Rightarrow **616 + 1$

$\Rightarrow *500 + 1$

$\Rightarrow C + 1$

$\Rightarrow D$

\therefore Output is: ELECTRICALD

5. (c)

char a[]="GATE2024";

char b[9];

strcpy(b, a); // The string a is copied to string b.

printf ("%s%d%d", b, strlen(b), sizeof(b));

//strlen(b)=8 and sizeof(b)=9

return 0;

Output: GATE202489

6. (a)

10	10	10	10	10	10	10	10	10	11
0	1	2	3	4	5	6	7	8	9
G	A	T	E	W	a	1	1	a	h

func("GATEWallah"); //Address of “GATEWallah” i.e 100 is passed.

ptr: 100

$*ptr$ or $*100 == G != 0$

printf ("%c", *ptr); //G is printed

func(102) is called. It prints *102 i.e T.

So, similarly, func(104), func(106), func(108), func(110) will be called.

Output: GTWla

7. (a)

10	10	10	10	10	10	10	10	10	11
0	1	2	3	4	5	6	7	8	9
G	A	T	E	W	a	1	1	a	h

p=100;

p+=4;//p=104

*104='\0'

printf("%s",p); //It will print from 104. 104 contains NULL.

Hence, no output.

8. (11)

10	10	10	10
0	1	2	3
G	A	T	E
20	20	20	20
0	1	2	3
P	a	r	a
30	30	30	30
0	1	2	3
W	a	l	1
			\0

40	40	40	40	40	40	40	40	40	40
0	1	2	3	4	5	6	7	8	
S	h	r	e	s	h	t	h	\0	

a:

50	50	50	51
0	4	8	2

10	20	30	40
0	0	0	0

b:

600	604	608	612
508	512	504	500

c	600 604 608
---	-------------

printf("%s", *--*++c+3);

//*--*++c+3 = *--*604+3 = *--512+3 = *508+3=300+3=303

//'lah' is printed.

printf("%s", **++c); //**608=*504=200

//'Parakram' is printed.

Output: lahParakram

Size of the output string: 11

Any issue with DPP, please report by clicking here:- <https://forms.gle/t2SzQVvQcs638c4r5>For more questions, kindly visit the library section: Link for web: <https://smart.link/sdfez8ejd80if>PW Mobile APP: <https://smart.link/7wwosivoicgd4>

Programming in C

Structures and Unions

DPP-01

[NAT]

```
1. #include <stdio.h>
union u{
    int a;
    char b;
    double d[2];
};
int main()
{
    union u u1;
    printf("%d", (int)sizeof(u1));
    return 0;
}
```

Assume that objects of the type int, char and double occupy 2 bytes, 1 bytes and 4 bytes, respectively.

The memory requirement for variable u1 is _____(in bytes).

[NAT]

2. Consider the following C declaration:

```
struct
{
    long a[3];
    union
    {
        int y;
        float z;
    }u;
} s;
```

Assume that objects of the type int, float and long occupy 2 bytes, 4 bytes and 8 bytes, respectively.

The memory requirement for variable s is _____(in bytes).

[MCQ]

```
3. #include <stdio.h>
struct s{
    char a, b;
};
void f(struct s *p){
    p->a+=2;
    p->b-=1;
}
int main()
{
    struct s s1, s2, *q;
    s1.a='A'; s1.b='C';
    q=&s1;
    f(q);
    printf("%c\t%c", s1.a, s1.b);
    return 0;
}
```

The output is:

- (a) C B
- (b) A C
- (c) Compilation error
- (d) Garbage values

[MCQ]

```
4. #include <stdio.h>
struct s{
    char a, b;
};
void f(struct s s1){
    s1.a+=3;
    s1.b-=1;
}
int main()
{
    struct s s1;
    s1.a='A'; s1.b='C';
    f(s1);
    printf("%c\t%c", s1.a, s1.b);
    return 0;
}
```

The output is:

- (a) C B
- (b) A C
- (c) Compilation error
- (d) Garbage values

[MCQ]

5. #include <stdio.h>

```
struct s{
    char a, b;
};

void f(struct s s1){
    s1.a+=32;
    s1.b+=32;
}

void g(struct s *p){
    static count=2;
    p->a+=count++;
    p->b+=++count;
}

int main()
{
    struct s s1, s2;
    s1.a='A'; s1.b='C';
    s2.a='B'; s2.b='D';
    f(s1);
    for(int i=0;i<2;i++)g(&s2);
    printf("%c\t%c",s1.a, s1.b);
    printf("\t%c\t%c",s2.a, s2.b);
    return 0;
}
```

The output is:

- (a) a c B D
- (b) A C B D
- (c) A C H N
- (d) a c B N

[MCQ]

6. #include <stdio.h>

```
struct days{
    char *q;
};

}s[]={ "Sunday", "Monday", "Tuesday", "Wednesday",
"Thursday", "Friday", "Saturday" };
```

```
int main()
```

```
{
    struct days *p=s;
    p=p+3;
    printf("%c", *p++>q);
    printf("%c", *++p>q);
    p=p-2;
    printf("%s",p>q);
    return 0;
}
```

The output string printed is-

- (a) WhWednesday
- (b) WTTuesday
- (c) WTWednesday
- (d) WhTuesday

[MSQ]

7. Which of the following statements are INCORRECT?

- (a) Functions cannot be defined inside the structure.
- (b) Structure variable of the same structure type can be defined inside a structure.
- (c) A function may not contain a structure defined in it.
- (d) Existing structure cannot be contained in another structure.

[NAT]

8. #include<stdio.h>

```
#include<string.h>
struct t
{
    char sname[20];
};
int main ()
{
    struct t t1, t2;
    strcpy(t1.sname, "GATEWallah"); //line a
    t2.sname="GATE2023"; //line b
    printf("%s", t1.sname); //line c
    printf("%s", t2.sname); //line d
    return 0;
}
```

The number of lines with error among lines a,b,c,d are _____.

Answer Key

- 1. (8)
- 2. (28)
- 3. (a)
- 4. (b)

- 5. (c)
- 6. (d)
- 7. (b, c, d)
- 8. (2)



Hints and Solutions

1. (8)

The size of the union is equal to the maximum size of the member variables

Here, double d[2] has the maximum size

$$\therefore \text{size of union} = (2 \times 4) \text{ bytes} = 8 \text{ bytes}$$

2. (28)

The size of the structure variable is the sum of the sizes of all its member variables

$$\begin{aligned}\text{Size of structure} &= \text{size of long } a[3] + \text{size of union} \\ &= 8 \times 3 + \max(4, 4) \\ &= 24 + 4 \\ &= 28\end{aligned}$$

3. (a)

a s1	b 'A' C 100
	'C' B 101

q [100] // q = &s1

f(100)

p [100]

100 → a += 2; // 100 → a = 'C'

100 → b -= 1; // 100 → b = 'B'

∴ printf() prints 'C B';

4. (b)

main()

a s1	b 'A' C 100
	'C' B 101

⇒ print("%c\t%c"; s1.a, s1.b);

⇒ A C

f(s1)

→ entire structure is passed

a s1	b 'A' D 100
	'C' B 101

s1.a += 3

s2.a -= 1

formal parameters are changed.

Actual parameters haven't changed

∴ Output: A C

5. (c)

a s1	b 'A' C 100
	101

a s2	b 'B' D H 200
	'D' H N 201

g(&s2); g(&s2);

f(s1)

→ entire structure is passed.

⇒ It will change the formal parameters but the actual parameters in main() won't change

g(200)

count	'Z' 'A' 4 200
	201

static

200 → a += 2; // D

200 → b += 4; // H

g(200)

count	'A' 'Z' 6 200
	201

200 → a += 4; // H

200 → b += 6; // N

∴ Output-

printf("%c\t%c", s1.a, s1.b);

printf("%c\t%c", s2.a, s2.b);

⇒ A C H N

6. (d)

100 101 102 103 104 105 106

S	u	n	d	a	y	\0
---	---	---	---	---	---	----

200 201 202 203 204 205 206

M	o	n	d	a	y	\0
---	---	---	---	---	---	----

300 301 302 303 304 305 306 307

T	u	e	s	d	a	y	\0
---	---	---	---	---	---	---	----

400 401 402 403 404 405 406 407 409 410

W	e	d	n	e	s	d	a	y	\0
---	---	---	---	---	---	---	---	---	----

500 501 502 503 504 505 506 507 508

T	h	u	r	s	d	a	y	\0
---	---	---	---	---	---	---	---	----

600 601 602 603 604 605 606

F	r	i	d	a	y	\0
---	---	---	---	---	---	----

700 701 702 703 704 705 706 707 708

S	a	t	u	r	d	a	y	\0
---	---	---	---	---	---	---	---	----

s [100 | 200 | 300 | 400 | 500 | 600 | 700]

↑

array of structures

p [800 | 812 | 816 | 808]

printf("%c", *p++ → q);

*812 → 400 ⇒ w

printf("%c", *++p → q);

*++812 → 500 ⇒ *++500

⇒ *++501

⇒ h

p=p-2;//p=808

printf("%s", p → q);

808 → 300 ⇒ Tuesday

∴ Output: WhTuesday

7. (b, c, d)

(a) CORRECT. Functions cannot be defined inside the structure

(b) INCORRECT. Structure variable of the same structure type cannot be defined inside structure.

(c) INCORRECT. A function can contain a structure defined in it.

(d) INCORRECT. Existing structure can be contained in another structure

8. (2)

line b: ERROR. Constant base address cannot be changed

line d: ERROR. As line 2 has error, line 4 cannot be executed.



Any issue with DPP, please report by clicking here:- <https://forms.gle/t2SzQVvQcs638c4r5>

For more questions, kindly visit the library section: Link for web: <https://smart.link/sdfez8ejd80if>



PW Mobile APP: <https://smart.link/7wwosivoicgd4>