**FA5**

1. Predict the output of the following code:

struct test

{

char str[20];

};

int main()

{

struct test s1,s2;

strcpy(s1.str,"seek");

s2=s1;

s1.str[0]='p';

printf("%s",s2.str);

return 0;

}

1. Segmentation fault
2. **seek**
3. peek
4. Compiler error
5. Presence of code like “s.t.b = 10” indicate:
6. Syntax error
7. **Structure**
8. Double data type
9. An ordinary variable name
10. Predict the output of the following C program:

#include <stdio.h>

struct test

{

int x,y,z;

};

int main()

{

struct test s1={.y=0,.z=1,.x=2};

printf("%d %d %d",s1.x,s1.y,s1.z);

return 0;

}

1. **2 0 1**
2. 0 1 2
3. 2 1 0
4. Compiler error
5. How to access the member of structure through pointer to a structure?

struct test

{

int x;

};

struct test s,\*p;

p=&s;

1. \*(p.x)
2. p->x
3. p.x
4. **both a and b**
5. Predict the output of the following code:

#include <stdio.h>

struct test

{

int x=0;

int y=0;

};

int main()

{

struct test s1={3,4};

printf("%d %d ",s1.x,s1.y);

return 0;

}

1. 0 0
2. garbage value
3. **error**
4. 3 4
5. Predict the output of the following code:

#include <stdio.h>

int main()

{

char \*a="formative assessment" ;

printf("%s",6+a);

return 0;

}

1. Error
2. **ive assessment**
3. format
4. ive
5. What would the following code snippet print:

#include <stdio.h>

int main()

{

char a[30]="formative/0assessment";

a[0]=a[21];

printf("%s",a);

return 0;

}

1. formative
2. **Nothing gets printed**
3. Garbage
4. Garbage followed by ormative assessment
5. What does the following fragment of C print?

#include <stdio.h>

int main()

{

char c[] = "FAC2018";

printf("%c",\*(c+(c[2]-c[1])));

return 0;

}

1. F
2. A
3. **C**
4. Invalid operation
5. Which of the following string declaration is valid:
6. char str[20];
7. char str[]=”abc”;
8. char \*p;
9. **All of the above**
10. If the two strings are identical, then **strcmp()** function returns
11. -1
12. 1
13. **0**
14. Yes
15. Which of the following function is more appropriate for reading in a multi-word string?
16. scanf()
17. **gets()**
18. getchar()
19. puts()
20. Which of the following function is correct that finds the length of a string?
21. **int xstrlen(char \*s)**

**{**

**int length=0;**

**while(\*s!='\0')**

**{**

**length++; s++;**

**}**

**return (length);**

**}**

1. int xstrlen(char s)

{

int length=0;

while(\*s!='\0')

length++; s++;

return (length);

}

1. int xstrlen(char \*s)

{

int length=0;

while(\*s!='\0')

length++;

return (length);

}

1. int xstrlen(char \*s)

{

int length=0;

while(\*s!='\0')

s++;

return (length);

}

1. What does the following declaration mean?

**int (\* arr) [12]**

1. Arr is array of pointers to 12 integers
2. **Arr is a pointer to an array of 12 integers**
3. Arr is an array of 10 integers
4. Arr is an pointer to array
5. Which of the following operation is not possible in C?

int a[5]={1,2,3,4,5};

int \*p=a;

1. p++
2. p=a
3. **a++**
4. p=a+1
5. What would be the output of following code snippet?

int arr[5]={3,1,5,4,2};

int \*ptr[5]={arr+2,arr,arr+4,arr+3,arr+1};

printf(“%d”,\*ptr[1]);

1. **3**
2. 1
3. 5
4. &arr[1]
5. Predict output of following program

#include <stdio.h>

int main()

{

int i;

intarr[5] = {1};

for (i = 0; i < 5; i++)

printf("%d ", arr[i]);

return 0;

}

1. 1 followed by four garbage values
2. **1 0 0 0 0**
3. 1 1 1 1 1
4. 0 0 0 0 0
5. What is the output of this C code?

#include <stdio.h>

void f(int a[2][])

{

a[0][1] = 3;

int i = 0, j = 0;

for (i = 0;i < 2; i++)

for (j = 0;j < 3; j++)

printf("%d", a[i][j]);

}

int main()

{

int a[2][3] = {0};

f(a);

return 0;

}

1. 0 3 0 0 0 0
2. 0 0 0 0 0 0
3. **Compile time error**
4. All garbage values
5. Comment on the 2 arrays regarding P and Q:

int \*a1[8];

int (\*a2)[8];

P. Array of pointers

Q. Pointer to an array

a**) p is a1, q is a2**

b) p is a2, q is a1

c) p is a2, q is a2

d) p is a1, q is a1

19. What is right way to Initialize array?

a) **int num [6];**

b) int num[ ];

c) int \*num={1,2,3};

d) all of the above

20. In an array all elements are stored in which of the following order in memory?

a) **sequential**

b) random

c) sequential and random

c) 1D in sequential and 2D in random