# C Programming lab Pointers quiz

**Name:** _____

**Email:** _____

**Roll nos:** _____

**Concept:** *Understanding pointers*

1. The below programming expression suggests what ?

   ```
   int *ptr;
   ```

   (a) unsigned integer
   (b) pointer to integer
   (c) integer
   (d) pointer to unsigned integer

2. The below programming expression suggests what ?

   ```
   int *ptr;
   ```

   (a) ptr contains unsigned integer value
   (b) ptr contains integer value
   (c) ptr contains address value

3. The below programming expression suggests what ?

   ```
   int *ptr[];
   ```

   (a) ptr is an array of pointers to integer
   (b) ptr is a pointer to array of integer

4. The below programming expression suggests what ?

   ```
   int *ptr();
   ```

   (a) ptr is a function that has no arguments and returns pointer to integer
   (b) ptr is a pointer to a function that has no arguments and returns integer

5. The below programming expression suggests what ?

   ```
   int (*ptr)();
   ```

   (a) ptr is a function that has no arguments and returns pointer to integer
   (b) ptr is a pointer to a function that has no arguments and returns integer

6. The below programming expression suggests what ?

   ```
   int (*ptr)(char);
   ```

   (a) ptr is a pointer to a function that has char type argument and returns integer

(b) ptr is a function that has one char type argument and returns pointer to integer

7. The below programming expression suggests what ?

```
int (*ptr)(char* arg);
```

(a) ptr is a pointer to a function that has char type argument and returns integer
(b) ptr is a function that has char type argument and returns pointer to integer
(c) ptr is a function that has pointer to char type argument and returns pointer to integer
(d) ptr is a pointer to a function that has pointer to char type argument and returns integer

8. The below programming expression suggests what ?

```
int (*ptr)(int *, int *);
```

(a) ptr is a function that has two integer type arguments and returns pointer to integer
(b) ptr is a function that has two pointer to integer type arguments and returns pointer to integer
(c) ptr is a pointer to a function that has two integer type arguments and returns integer
(d) ptr is a pointer to a function that has two pointer to integer type arguments and returns integer

9. The below programming expression suggests what ?

```
int (*ptr)(int (*arg)());
```

(a) ptr is a function that has an integer type argument and returns pointer to integer
(b) ptr is a function that has pointer to a function argument and returns pointer to integer
(c) ptr is a function pointer that takes function pointer as an argument and returns integer
(d) ptr is a function that has pointer to integer type argument and returns pointer to integer

10. The below programming expression suggests what ?

```
int (*ptr)(int (*arg)(int a));
```

(a) ptr is a pointer to a function that has pointer to a function argument which has an integer argument and returns integer
(b) ptr is a pointer to a function that has pointer to integer type argument and returns pointer to integer
(c) ptr is a pointer to a function that has pointer to a function argument and returns pointer to integer
(d) ptr is a pointer to a function that has an integer type argument and returns pointer to integer
(e) ptr is a pointer to a function that has pointer to a function argument which has an integer argument and returns pointer to integer

11. The below programming expression suggests what ?

```
int (*ptr)(int (*arg)(), char);
```

(a) ptr is a pointer to a function that has an argument of function pointer and an char type argument and returns integer
(b) ptr is a pointer to a function that has an argument of function pointer and returns pointer to integer

12. The below programming expression suggests what ?

```
int *ptr(int (*arg)(), char);
```

(a) ptr is a function that has a function pointer argument and char type argument and returns pointer to integer

(b) ptr is a pointer to a function that has a function pointer argument and char type argument and returns integer

13. The below programming expression suggests what ?

```
int *ptr[];
```

(a) ptr is a pointer to an array of integer
(b) ptr is a pointer to a function which returns integer
(c) ptr is a function which returns pointer to integer
(d) ptr is a array of pointers to integer

14. The below programming expression suggests what ?

```
int (*ptr)[];
```

(a) ptr is a array of pointers to integers
(b) ptr is a function which returns pointer to integer
(c) ptr is a pointer to a function which returns integer
(d) ptr is a pointer to an array of integers

15. The below programming expression suggests what ?

```
int *(*ptr)[];
```

(a) ptr is a array of pointers to integer
(b) ptr is a pointer to an array of integer pointers
(c) ptr is a function to an arry of integer pointers
(d) ptr is a pointer to an array of integer

16. The below programming expression suggests what ?

```
int (*ptr[])();
```

(a) ptr is a array of function pointers that takes function pointer as an argument and returns pointer to an integer
(b) ptr is a array of function pointers that takes function pointer as an argument and returns integer
(c) ptr is a array of pointers to integer
(d) ptr is a array of function pointers that takes no argument and returns integer

17. The below programming expression suggests what ?

```
int (*ptr[])(int (*arg)());
```

(a) ptr is a array of function pointers that takes function pointer as an argument and returns pointer to an integer
(b) ptr is a array of pointers to integer
(c) ptr is a array of function pointers that takes function pointer as an argument and returns integer
(d) ptr is a array of function pointers that takes no argument and returns integer

18. The below programming expression suggests what ?

```
int* (*ptr[])(int (*arg)());
```

(a) ptr is a array of function pointers that takes function pointer as an argument and returns pointer to an integer

(b) ptr is a array of function pointers that takes no argument and returns integer

(c) ptr is a array of function pointers that takes function pointer as an argument and returns integer

(d) ptr is a array of pointers to integer

**Concept:** *indirection operator*

19. In pointers, num[i] also means *(num+i)

    (a) No

    (b) Yes

20. In pointers, num[i] also means (num+i)

    (a) No

    (b) Yes

21. In pointers, num[i] also means &(num+i)

    (a) No

    (b) Yes

22. In pointers, num[i][j] also means *(num+i+j)

    (a) No

    (b) Yes

23. In pointers, num[i][j] also means *(num[i]+j) ?

    (a) No

    (b) Yes

24. In pointers, num[i][j] also means *(*(num+i)+j)

    (a) No

    (b) Yes

25. In pointers, num[i][j] also means *(&(num+i)+j)

    (a) Yes

    (b) No

26. In pointers, num[i][j] also means &(&(num+i)+j)

    (a) Yes

    (b) No

27. In pointers, num[i][j] also means &(&(num+i)*j)

    (a) No

    (b) Yes

**Concept:** *address value of pointer*

28. In the following section of code, what is the output ?

```
int data[2][2] = { { 1,2 },{ 3,4 } };
int (*p)[2];
p = &data[0];
printf("%p",p);
```

(a) Address of data[0][1]

(b) Address of data[0][0]

(c) Address of data[1][0]

(d) Address of data[1][1]

(e) garbage value

29. In the following section of code, what is the output ?

```
int data[2][2] = { { 1,2 },{ 3,4 } };
int (*p)[2];
p = &data[1];
printf("%p",p);
```

(a) Address of data[0][0]

(b) garbage value

(c) Address of data[0][1]

(d) Address of data[1][0]

(e) Address of data[1][1]

30. In the following section of code, what is the output ?

```
int data[2][2] = { { 1,2 },{ 3,4 } };
int (*p)[2];
p = &data[1];
printf("%p",p-1);
```

(a) Address of data[1][1]

(b) Address of data[0][1]

(c) garbage value

(d) Address of data[0][0]

(e) Address of data[1][0]

31. In the following section of code, what is the output ?

```
int data[2][2] = { { 1,2 },{ 3,4 } };
int (*p)[2];
p = &data[0];
printf("%p",p+1);
```

(a) Address of data[0][0]

(b) Address of data[1][0]

(c) Address of data[1][1]

(d) Address of data[0][1]

(e) garbage value

32. In the following section of code, what is the output ?

```
int data[2][2] = { { 1,2 },{ 3,4 } };
int *p[2];
p[0] = data[0];
printf("%p",p[0]);
```

(a) garbage value

(b) Address of data[1][0]

(c) Address of data[1][1]

(d) Address of data[0][1]

(e) Address of data[0][0]

33. In the following section of code, what is the output ?

```
int data[2][2] = { { 1,2 },{ 3,4 } };
int *p[2];
p[1] = data[0];
printf("%p",p[0]);
```

(a) Address of data[0][1]

(b) Address of data[0][0]

(c) Address of data[1][0]

(d) garbage value

(e) Address of data[1][1]

34. In the following section of code, what is the output ?

```
int data[2][2] = { { 1,2 },{ 3,4 } };
int *p[2];
p[0] = data[0];
printf("%d",p[0][0]);
```

35. In the following section of code, what is the output ?

```
int data[2][2] = { { 1,2 },{ 3,4 } };
int *p[2];
p[0] = data[0];
printf("%d",p[0][1]);
```

36. In the following section of code, what is the output ?

```
int data[2][2] = { { 1,2 },{ 3,4 } };
int *p[2];
p[0] = data[0];
printf("%d",p[1][0]);
```

37. In the following section of code, what is the output ?

```
int data[2][2] = { { 1,2 },{ 3,4 } };
int *p[2];
p[0] = data[0];
printf("%d",p[1][1]);
```

**Concept:** *Initializing pointers*

38. Is this set of expressions valid ?

```
int x = 10;
int *ptr=&x;
```

(a) Yes

(b) No

39. Is this set of expressions valid ?

```
int x = 10;
int *ptr=x;
```

(a) No

(b) Yes

40. Is this set of expressions valid ?

```
int x = 97;
int *ptr=&x;
printf("%c",*ptr);
```

(a) No

(b) Yes

41. Is this set of expressions valid ?

```
int x = 97;
int *ptr=&x;
printf("%f",*ptr);
```

(a) Yes

(b) No

42. Is this set of expressions valid ?

```
int x = 97;
int *ptr=&x;
printf("%f",(float)*ptr);
```

(a) No

(b) Yes

43. Is this set of expressions valid ?

```
int x = 97;
int *ptr=&x;
x = 120;
printf("%c",*ptr);
```

(a) No

(b) Yes

44. Is this set of expressions valid ?

```
int x = 97;
int *ptr=&x;
*ptr = 120;
printf("%c",*ptr);
```

(a) No

(b) Yes

45. Is this set of expressions valid ?

```
int x = 97;
char *ptr;
ptr = (char*)&x;
```

(a) Yes

(b) No

46. What is the output of this program ?

```
int x = 97;
int *ptr=&x;
*ptr = 120;
printf("%c",*ptr);
```

(a) 'y'

(b) 'x'

(c) 'b'

(d) 'a'

47. What is the output of this program ?

```
int x = 97;
int *ptr=&x;
*ptr = 120;
printf("%c",x);
```

(a) 'y'

(b) 'x'

(c) 'a'

(d) 'b'

48. What is the output of this program ?

```
int x = 97;
int *ptr=&x;
x = 120;
printf("%c",x);
```

(a) 'a'

(b) 'x'

(c) 'b'

(d) 'y'

49. What is the output of this program ?

```
int x = 97;
int *ptr=&x;
x = 120;
printf("%c",*ptr);
```

(a) 'b'

(b) 'a'

(c) 'y'

(d) 'x'

50. What is the output of this program ?

```
int x = 97;
int *ptr;
*ptr = x;
x = 120;
printf("%c",x);
```

(a) 'x'

(b) 'a'

(c) 'b'

(d) 'y'

51. What is the output of this program ?

```
int x = 97;
int *ptr;
*ptr = x;
x = 120;
printf("%c",*ptr);
```

(a) 'a'

(b) 'x'

(c) 'b'

(d) 'y'

52. What is the output of this program ?

```
int x = 97;
char *ptr;
ptr = (char*)&x;
printf("%c",*ptr);
```

(a) 'a'

(b) 'x'

(c) 'y'

(d) 'b'

53. What is the output of this program ?

```
int x = 97;
char *ptr;
ptr = (char*)&x;
printf("%d",*ptr);
```

(a) 'a'

(b) 'b'

(c) 98

(d) 97

**Concept:** *Passing 2-D array using pointers*

54. What is the output of the following program ?

```
#include<stdio.h>
void test(int *ptr, int r, int c)
    {
    int i,j;
    for(i=0;i<r;i++)
        {
        for(j=0;j<c;j++)
            printf("%d ", *(ptr+i*c+j));
        printf("\n");
        }
    }

int main()
    {
    int data[2][2]={1,2,3,4};
    test(&data[0][0],2,2);
    }
```

55. What is the output of the following program ?

```
#include<stdio.h>

void test(int *ptr, int r, int c)
    {
    int i,j;
    for(i=0;i<r;i++)
        {
        for(j=0;j<c;j++)
            printf("%d ", ptr[i][j]);
        printf("\n");
        }
    }

int main()
    {
    int data[2][2]={1,2,3,4};
    test(&data[0][0],2,2);
    }
```

56. What is the output of the following program ?

```
#include<stdio.h>

void test(int *ptr, int r, int c)
    {
    int i,j;
    for(i=0;i<r;i++)
        {
        for(j=0;j<c;j++)
            printf("%d ",*(ptr[i]+j));
        printf("\n");
        }
    }

int main()
    {
    int data[2][2]={1,2,3,4};
    test(&data[0][0],2,2);
    }
```

57. What is the output of the following program ?

```
#include<stdio.h>

void test(int *ptr, int r, int c)
    {
    int i,j;
    for(i=0;i<r;i++)
        {
        for(j=0;j<c;j++)
            printf("%d ",*(&ptr[i]+j));
        printf("\n");
        }
    }
int main()
    {
    int data[2][2]={1,2,3,4};
    test(&data[0][0],2,2);
    }
```

58. What is the output of the following program ?

```
#include<stdio.h>

void test(int *ptr, int r, int c)
    {
    int i,j;
    for(i=0;i<r;i++)
```

```
        {
        for(j=0;j<c;j++)
            printf("%d ",*(ptr+j*c+i));
        printf("\n");
        }
    }
int main()
    {
    int data[2][2]={1,2,3,4};
    test(&data[0][0],2,2);
    }
```

59. What is the output of the following program ?

```
#include<stdio.h>

void test(int (*ptr)[], int r, int c)
    {
    int i,j;
    for(i=0;i<r;i++)
        {
        for(j=0;j<c;j++)
            printf("%d ",ptr[i][j]);
        printf("\n");
        }
    }
int main()
{
    int data[2][2]={1,2,3,4};
    test(data,2,2);
}
```

60. What is the output of the following program ?

```
#include<stdio.h>

void test(int (*ptr)[2], int r, int c)
    {
    int i,j;
    for(i=0;i<r;i++)
        {
        for(j=0;j<c;j++)
            printf("%d ",ptr[i][j]);
        printf("\n");
        }
    }
int main()
```

```
    {
    int data[2][2]={1,2,3,4};
    test(data,2,2);
    }
```

61. What is the output of the following program ?

```
#include<stdio.h>

void test(int (*ptr)[2], int r, int c)
    {
    int i,j;
    for(i=0;i<r;i++)
        {
        for(j=0;j<c;j++)
            printf("%d ",ptr[j][i]);
        printf("\n");
        }
    }
int main()
    {
    int data[2][2]={1,2,3,4};
    test(data,2,2);
    }
```

62. What is the output of the following program ?

```
#include<stdio.h>

void test(int ptr[][2], int r, int c)
    {
    int i,j;
    for(i=0;i<r;i++)
        {
        for(j=0;j<c;j++)
            printf("%d ",ptr[i][j]);
        printf("\n");
        }
    }
int main()
    {
    int data[2][2]={1,2,3,4};
    test(data,2,2);
    }
```

63. What is the output of the following program ?

```
#include<stdio.h>

void test(int **ptr, int r, int c)
    {
    int i,j;
    for(i=0;i<r;i++)
        {
        for(j=0;j<c;j++)
            printf("%d ",ptr[i][j]);
        printf("\n");
        }
    }
int main()
    {
    int data[2][2]={1,2,3,4}, **ptr, *temp;
    temp = &data[0][0];
    ptr = &temp;
    test(ptr,2,2);
    }
```

64. What is the output of the following program ?

```
#include<stdio.h>

void test(int **ptr, int r, int c)
    {
    int i,j;
    for(i=0;i<r;i++)
        {
        for(j=0;j<c;j++)
            printf("%d ",ptr[i][j]);
        printf("\n");
        }
    }
int main()
    {
    int data[2][2]={1,2,3,4}, **ptr, *temp[2];
    temp[0] = &data[0][0];
    temp[1] = &data[1][0];
    ptr = &temp[0];
    test(ptr,2,2);
    }
```

65. What is the output of the following program ?

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

void copy(int **p, int **d, int r, int c)
    {
    int i,j;
    d = (int**)malloc(sizeof(int*)*r);
    for(i=0;i<r;i++)
        {
        d[i] = (int*)malloc(sizeof(int)*c);
        }

    for(i=0;i<r;i++)
        for(j=0;j<c;j++)
            d[i][j] = p[i][j];

    }



void printBoard(int **d, int r, int c)
    {
    int i,j;
    for(i=0;i<r;i++)
        {
        for(j=0;j<c;j++)
            printf("%d ",d[i][j]);
        printf("\n");
        }
    }

int main()
    {
    int data[2][2]={1,2,3,4}, **ptr, *temp[2],**dest;
    temp[0] = &data[0][0];
    temp[1] = &data[1][0];
    ptr = &temp[0];
    copy(ptr,dest,2,2);
    printBoard(dest,2,2);
    free(dest);
    }
```

66. What is the output of the following program ?

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

int** copy(int **p, int r, int c)
    {
    int i,j,**d;
    d = (int**)malloc(sizeof(int*)*r);
    for(i=0;i<r;i++)
        {
        d[i] = (int*)malloc(sizeof(int)*c);
        }

    for(i=0;i<r;i++)
        for(j=0;j<c;j++)
            d[i][j] = p[i][j];

    return d;

    }

void printBoard(int **d, int r, int c)
    {
    int i,j;

    for(i=0;i<r;i++)
        {
        for(j=0;j<c;j++)
            printf("%d ",d[i][j]);
        printf("\n");
        }
    }

int main()
    {
    int data[2][2]={1,2,3,4}, **ptr, *temp[2],**dest;
    temp[0] = &data[0][0];
    temp[1] = &data[1][0];
    ptr = &temp[0];
    dest = copy(ptr,2,2);
    printBoard(dest,2,2);
    free(dest);
    }
```

**Concept:** *Writing programs*

67. Write a function which will determine maximum and minimum number of all elements in a 2-D array ?

```
Your program
```

68. Write a function which will obtain the transpose of a 4 X 4 matrix ?

    `Your Program`

69. Write a function to sort all the elements of a 3 X 3 matrix ?

    `Your Program`

70. Write a function to add two 6 X 6 matrices ?

    `Your Program`

71. Write a function to multiply two 6 X 6 matrices ?

```
Your Program
```

72. Write a function to check whether the square matrix is symmetric ?

```
Your Program
```