Program 1:

```
#include <stdio.h>

int main() {
    int arr[] = {1, 2, 3, 4, 5};
    int *ptr = &arr[0];
    printf("%d\n", *++ptr);
    printf("%d\n", *(ptr++));
    return 0;
}
```

View output

2

3

Program 2:

```
#include <stdio.h>

int main() {
    int i = 0;
    for (; i < 5; i++) {
        if (i == 2)
            continue;
        printf("%d ", i);
    }
    return 0;
}
```

View output

0 1 3 4

Program 03:

```
#include <stdio.h>

int main() {
    int x = 5;
    int *ptr = &x;
    printf("%d %p\n", *ptr, ptr);
    (*ptr)++;
    printf("%d\n", x);
    return 0;
}
```

View output

5 address of x

6

Program 04:

```c
#include <stdio.h>

int main() {
    int x = 10;
    int *ptr1 = &x, *ptr2 = &x;
    (*ptr1)++;
    printf("%d %d\n", *ptr1, *ptr2);
    return 0;
}
```

View output

11 11

Program 05:

```c
#include <stdio.h>

int main() {
    char str[] = "Hello, World!";
    char *ptr = str;
    printf("%c\n", *ptr++);
    printf("%s\n", ptr);
    return 0;
}
```

View output

```
H
ello, World!
```

Program 06:

```c
#include <stdio.h>

int main() {
    int x = 5;
    printf("%d\n", x++);
    printf("%d\n", ++x);
    return 0;
}
```

View output

```
5
7
```

Program 07:

```c
#include <stdio.h>

int main() {
    int a = 10, b = 20;
    printf("%d\n", a++ + b++);
    printf("%d %d\n", a, b);
    return 0;
}
```

View output

```
30

11 21
```

Program 08:

```c
#include <stdio.h>

int main() {
    int i = 10;
    while (i++ <= 15) {
        printf("%d ", i);
    }
    return 0;
}
```

View output

```
11 12 13 14 15 16
```

Program 09:

```c
#include <stdio.h>

int main() {
    int arr[] = {1, 2, 3, 4, 5};
    int *ptr = arr;
    printf("%d\n", *ptr++);
    printf("%d\n", *ptr);
    return 0;
}
```

View output

```
1
2
```

Program 10:

```
#include <stdio.h>

int main() {
    int x = 10, y = 20;
    int *ptr1 = &x, *ptr2 = &y;
    printf("%d %d\n", *ptr1, *ptr2);
    ptr1 = ptr2;
    printf("%d %d\n", *ptr1, *ptr2);
    return 0;
}
```

View output

```
10 20
20 20
```

Program 11:

```
#include <stdio.h>

void func(int *a, int *b) {
    int temp = *a;
    *a = *b;
    *b = temp;
}

int main() {
    int x = 10, y = 20;
    printf("Before swapping: x = %d, y = %d\n", x, y);
    func(&x, &y);
    printf("After swapping: x = %d, y = %d\n", x, y);
    return 0;
}
```

View output

Before swapping: x = 10, y = 20

After swapping: x = 20, y = 10

Program 12:

```
#include <stdio.h>

int main() {
    char *str = "Hello";
    printf("%c\n", *str);
    printf("%s\n", str);
    return 0;
}
```

View output

H

Hello

Program 13:

```
#include <stdio.h>

int main() {
    int a = 10;
    int *ptr = &a;
    printf("%d\n", *ptr);
    *ptr = 20;
    printf("%d\n", a);
    return 0;
}
```

View output

10

20

Program 14:

```
#include <stdio.h>

int main() {
    int x = 10, y = 20;
    int *ptr1 = &x, *ptr2 = &y;
    *ptr1 += *ptr2;
    printf("%d\n", x);
    return 0;
}
```

View output

30

Program 15:

```
#include <stdio.h>

int main() {
    int arr[] = {1, 2, 3, 4, 5};
    printf("%d\n", *arr + 3);
    return 0;
}
```

View output

4

Program 16:

```
#include <stdio.h>

void swap(int *a, int *b) {
    int temp = *a;
    *a = *b;
    *b = temp;
}

int main() {
    int x = 5, y = 10;
    swap(&x, &y);
    printf("x = %d, y = %d\n", x, y);
    return 0;
}
```

View output

x = 10, y = 5

Program 17:

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

```
int main() {
    int arr[] = {1, 2, 3, 4, 5};
    int *ptr = arr;
    printf("%d\n", *ptr++);
    printf("%d\n", *ptr);
    return 0;
}
```

View output

1

2

Program 18:

```
#include <stdio.h>

int main() {
    int i = 0;
    while (i < 5) {
        printf("%d ", ++i);
    }
    return 0;
}
```

View output

1 2 3 4 5

Program 19:

```
#include <stdio.h>

int main() {
    char str[] = "Hello";
    char *ptr = str;
    printf("%c\n", *(ptr + 1));
    printf("%s\n", ptr + 2);
    return 0;
}
```

View output

```
e
llo
```

Program 20:

```
#include <stdio.h>

int main() {
    int arr[] = {1, 2, 3, 4, 5};
    printf("%d\n", *arr + 2);
    return 0;
}
```

View output

3

Program 21:

```
#include <stdio.h>

int main() {
    int arr[] = {10, 20, 30, 40, 50};
    int *ptr = arr;
    printf("%d\n", *ptr++);
    printf("%d\n", *++ptr);
    return 0;
}
```

View output

10

30

Program 22:

```c
#include <stdio.h>

int main() {
    int x = 10, y = 5;
    int *ptr1 = &x, *ptr2 = &y;
    *ptr1 = *ptr1 + *ptr2;
    printf("%d\n", x);
    return 0;
}
```

View output

15

Related posts:

1. Relationship among entities
2. Introduction of IOT
3. Marketing Managment RGPV Diploma Paper Solved
4. Value of function in programming
5. Hardware components and device solved paper RGPV Diploma
6. USE CASE for MCQ application
7. OS Interview Q & A | Part 01 | Prof. Jayesh Umre
8. Compilation
9. OOPs in C# | PPL | Prof. Jayesh Umre

10. Overloaded subprograms
11. Static and Dynamic scope
12. Type Checking
13. Testing Levels | Software engineering | SEPM | Prof. Jayesh Umre
14. Static and Dynamic Analysis | Software Engineering| SEPM| Prof. Jayesh Umre
15. Code Inspection | Software engineering | SEPM | Prof. Jayesh Umre
16. Code Inspection
17. Characterstics of IOT
18. IOT Internet of Things
19. Monitors
20. Static and Stack-Based Storage management
21. Message passing
22. Exception handler in Java
23. Exception Propagation
24. Concept of Binding
25. Data mining and Data Warehousing
26. Introduction to Concurrency Control
27. Introduction to Transaction
28. Introduction to Data Models
29. Coaxial Cable
30. DHCP
31. DNS
32. Introduction to SNMP
33. Introduciton to SMTP
34. Introduction to NFS
35. Introduction to Telnet
36. Introduction to FTP

37. Internet Intranet Extranet
38. UGC NET Notes
39. Computer Terminologies
40. UGC NET Paper 1 December 2012
41. UGC Net paper 1 June 2011
42. closure properties of regular languages
43. Functional programming languages
44. Virtualization fundamental concept of compute
45. Dia software for UML, ER, Flow Chart etc
46. DAVV MBA: Business Communication
47. Mirroring and Striping
48. RGPV Solved Papers
49. CD#08 | Semantic analysis phase of compiler in Hindi video | Semantic tree | Symbol table | int to real
50. COA#27 | Explain the Memory Hierarchy in short. | COA previoys years in Hindi video
51. Infix to Postfix expression
52. Array implementation of Stack
53. Stack Data Structure
54. DBMS#03 | DBMS System Architecture in Hindi video
55. Java program method overloaing
56. Java program use of String
57. DS#33 | 2 Dimensional Array | Data Structure in Hindi video
58. SE#10 | Function point (FP) project size estimation metric in Hindi video
59. ADA#02 | Define Algorithm. Discuss how to analyse Algorithm | ADA previous years in Hindi video
60. Principles of Programming Languages
61. Discrete Structures

62. Machine Learning
63. R Programming Video Lectures
64. Internet of Things (IOT)
65. Digital Circuits
66. Number Systems
67. Computer Organization and Architecture Video Lectures
68. UGC NET
69. There are five bags each containing identical sets of ten distinct chocolates. One chocolate is picked from each bag.The probability that at least two chocolates are identical is ______
70. C Programming Questions
71. What is Software ? What is the difference between a software process and a software product ?
72. Difference between scopus and sci/scie journal
73. Human Process Interventions: Individual and Group Level & Organization Level Topics Covered: Coaching, training and development, conflict resolution process process consultation, third-party interventions, and team building.
74. Leading and Managing Change & Emerging Trends in OD
75. Designing and Evaluating Organization Development Interventions
76. Tutorial
77. Data Dictionary and Dynamic Performance Views
78. Anna University Notes | Big Data Analytics
79. What is Map Reduce programming model? Explain.
80. Features of Web 2.0
81. Describe in brief the different sources of water.
82. RGPV BEEE
83. Define data structure. Describe about its need and types.Why do we need a data type

?

84. Interview Tips