

Section-A:2

1. Find the largest and smallest of two values.

Code: #include <stdio.h>

```
int main() {
    int num1, num2;
    printf("Enter the first number:");
    scanf("%d", &num1);
    printf("Enter the second number:");
    scanf("%d", &num2);
    if (num1 > num2) {
        printf("The largest number is: %d\n", num1);
        printf("The smallest number is: %d\n", num2);
    } else if (num2 > num1) {
        printf("The largest number is: %d\n", num2);
        printf("The smallest number is: %d\n", num1);
    } else {
        printf("Both numbers are equal: %d\n", num1);
    }
    return 0;
}
```

Output: Enter the first number: 1
Enter the second number: 7
The largest number is: 7
The smallest number is: 1

2. Find the largest and smallest of three values.

Code: #include <stdio.h>

```
int main() {
    int num1, num2, num3;
    int largest, smallest;
    printf("Enter three numbers:");
    scanf("%d %d %d", &num1, &num2, &num3);
    largest = num1;
    if (num2 > largest) {
        largest = num2;
    }
    if (num3 > largest) {
        largest = num3;
    }
    smallest = num1;
    if (num2 < smallest) {
        smallest = num2;
    }
    if (num3 < smallest) {
        smallest = num3;
    }
    printf("Largest number: %d\n", largest);
    printf("Smallest number : %d\n", smallest);
    return 0;
}
```

Output: Enter three numbers: 2 4 7

Largest number: 7

smallest number: 2

3. Calculate net salary where:

$$\text{Net Salary} = \text{Gross Salary} + \text{Allowances} - \text{Deductions}$$

Code: #include <stdio.h>

```
int main() {
    float grossSalary;
    float allowance;
    float deduction;
    float netSalary;
    printf("Enter Gross Salary:");
    scanf("%f", &grossSalary);
    if (grossSalary > 10000) {
        allowance = 0.10 * grossSalary;
        deduction = 0.03 * grossSalary;
    } else if (grossSalary > 5000) {
        allowance = 0.07 * grossSalary;
        deduction = 0.02 * grossSalary;
    } else {
        allowance = 0;
        deduction = 0;
    }
    netSalary = grossSalary + allowance - deduction;
    printf("Net Salary: %.2f\n", netSalary);
    return 0;
}
```

Output: Enter Gross Salary: 6000

Net Salary: 6300

4. Check if number is divisible by 7 or not.

Code: #include <stdio.h>

```
int main() {
    int number;
    printf("Enter an integer:");
    scanf("%d", &number);
    if (number % 7 == 0) {
        printf("%d is divisible by 7.\n", number);
    } else {
        printf("%d is not divisible by 7.\n", number);
    }
    return 0;
}
```

Output: Enter an integer: 49

49 is divisible by 7.

5. Calculate net sales:

$$\text{Net sales} = \text{Gross Sales} - \text{Discount}$$

Code: # include <stdio.h>

```
int main() {
    float grossSales;
    float discountRate;
    float netSales;
    printf("Enter Gross Sales:");
    scanf("%f", &grossSales);
    if (grossSales > 20000) {
        discountRate = 0.15;
    } else if (grossSales > 10000) {
        discountRate = 0.10;
    } else {
        discountRate = 0.05;
    }
    netSales = grossSales - (grossSales * discountRate);
    printf("Net Sales: %.2f\n", netSales);
    return 0;
}
```

Output: Enter Gross Sales : 60000

Net Sales : 51000.00

6. Calculate total and average of three subjects and assign grades.

Code : #include <stdio.h>

```
int main() {
    int subject1, subject2, subject3;
    int total;
    float average;
    printf("Enter marks for subject 1: ");
    scanf("%d", &subject1);
    printf("Enter marks for subject 2: ");
    scanf("%d", &subject2);
    printf("Enter marks for subject 3: ");
    scanf("%d", &subject3);
    total = subject1 + subject2 + subject3;
    average = (float) total / 3;
    printf("\nTotal Marks: %d\n", total);
    printf("Average Marks: %.2f\n", average);
    if (subject1 < 35 || subject2 < 35 || subject3 < 35) {
        printf("Grade: Fail (Failed in one or more subjects)\n");
    } else if (average >= 70) {
        printf("Grade: Distinction\n");
    } else if (average >= 60) {
        printf("Grade: First class\n");
    } else if (average >= 50) {
        printf("Grade: Second class\n");
    } else if (average >= 35) {
        printf("Grade: Third class\n");
    } else {
        printf("Grade: Fail\n");
    }
    return 0;
}
```

Output: Enter marks for subject 1: 99
Enter marks for subject 2: 87
Enter marks for subject 3: 88
Total Marks: 274
Average Marks: 91.33
Grade: Distinction.

1. Print first 10 natural numbers.

Ans:
code: #include <stdio.h>
int main() {
 int m;
 for (m=1; m<=10; m++) {
 printf ("%d\n", m);
 }
 return 0;
}

Output: 1
2
3
4
5
6
7
8
9
10

2. Print first 10 odd numbers.

Ans:
code: #include <stdio.h>
int main() {
 int j;
 printf ("The first 10 odd numbers are:\n");
 for (j=0; j<10; j++) {
 printf ("%d", (2*j) + 1);
 }
 printf ("\n");
 return 0;
}

Output: 1
3
5
7
9
11
13
15
17
19

3. Print first 10 even numbers.

Ans:
code: #include <stdio.h>
int main() {
 int i;
 int count = 0;
 printf ("First 10 even numbers are:");
 for (i=2; count < 10; i+=2) {
 printf ("%d", i);
 count++;
 }
 printf ("\n");
 return 0;
}

Output: 2
4
6
8
10
12
14
16
18
20

4. Print first n natural numbers.

code: #include <stdio.h>
int main() {
 int n, i;
 printf ("Enter the value of n:");
 scanf ("%d", &n);
 printf ("First %d natural numbers are:\n", n);
 for (i=1; i<=n; i++) {
 printf ("%d\n", i);
 }
 return 0;
}

Output: Enter the value of n: 7

First 7 natural numbers are:

1
2
3
4
5
6
7

5. Print first n odd numbers.

code: #include <stdio.h>
int main() {
 int n, i;
 printf ("Enter the number of odd numbers to print:");
 scanf ("%d", &n);
 printf ("First %d odd numbers are:\n", n);
 for (i=0, i<n; i++) {
 printf ("%d", 2*i+1);
 }
 printf ("\n");
 return 0;
}

Output: Enter the number of odd numbers to print: 2

First 2 odd numbers are: 1 3

6. Print first n even numbers.

Code: # include <stdio.h>

```
int main() {
    int n, i;
    printf ("Enter the number of even numbers to print : ");
    scanf ("%d", &n);
    if (n<=0) {
        printf ("Please enter a positive number.\n");
        return 1;
    }
    printf ("First %d even numbers are :\n", n);
    for (i=1; i<=n; i++) {
        printf ("%d\n", 2*i);
    }
    return 0;
}
```

Output: Enter the number of even numbers to print : 3

First 3 even numbers are :

2
4
6

7. Print sum of n natural numbers.

Code: # include <stdio.h>

```
int main() {
    int n, sum=0, i;
    printf ("Enter a positive integer (n) : ");
    scanf ("%d", &n);
    if (n<1) {
        printf ("Please enter a positive integer.\n");
        return 1;
    }
    for (i=1; i<=n; i++) {
        sum+=i;
    }
    printf ("Sum of first %d natural numbers is : %d\n", n, sum);
    return 0;
}
```

Output: Enter a positive integer (n) : 3

Sum of first 3 natural numbers is : 6

8. Print sum of first n odd numbers.

Code : #include <stdio.h>

```
int main() {
    int n, sum = 0, i;
    printf ("Enter the number of odd numbers(n):");
    scanf ("%d", &n);
    for (i=1; i<=n; i++) {
        sum += (2*i - 1);
    }
    printf ("Sum of first %d odd numbers is: %d\n", n, sum);
    return 0;
}
```

Output : Enter the number of odd numbers(n): 7

Sum of first 7 odd numbers is : 49.

9. Print sum of first n even numbers.

Code : # include <stdio.h>

```
int main() {
    int n, sum = 0, i;
    printf ("Enter the value of n:");
    scanf ("%d", &n);
    for (i=1; i<=n; i++) {
        sum += (2*i);
    }
    printf ("Sum of first %d even numbers is: %d\n", n, sum);
    return 0;
}
```

Output : Enter the value of n: 5

Sum of first 5 even numbers is : 30

20. Print factorial of a number. (S1=120)

Code:

```
#include <stdio.h>
int main() {
    int n, i;
    long long factorial = 1;
    printf ("Enter an integer:");
    scanf ("%d", &n);
    if (n<0) {
        printf ("Factorial of a negative number doesn't exist.\n");
    } else {
        for (i=1; i<=n; i++) {
            factorial *= i;
        }
        printf ("Factorial of %d = %lld\n", n, factorial);
    }
    return 0;
}
```

Output:

```
Enter an integer = 5
Factorial of 5 = 120
```

21. Print your name 5 times.

Code:

```
#include <stdio.h>
int main () {
    for (int i=0; i<5; i++) {
        printf ("Danshan\n");
    }
    return 0;
}
```

Output:

```
Danshan
Danshan
Danshan
Danshan
Danshan
```

22. Print your name n times.

Code: #include <stdio.h>
int main() {
 int n,i;
 printf ("Enter the number of times to print your name:");
 scanf ("%d", &n);
 for (i=0 ; i < n ; i++) {
 printf (" Darshan\n");
 }
 return 0;
}

Output: Enter the number of times to print your name: 2

Darshan

Darshan

13. Print sum of numbers divisible by 13 from 1 to 100 numbers.

Code: #include <stdio.h>

```
int main() {
    int sum = 0;
    int i;
    for (i=1; i<=100; i++) {
        if (i%13 == 0) {
            sum += i;
        }
    }
    printf ("The sum of numbers divisible by 13 from 1 to 100
            is : %d\n", sum);
    return 0;
}
```

Output: The sum of numbers divisible by 13 from 1 to 100 is: 364.

14. Calculate sum and mean of 10 values.

Code: #include <stdio.h>

```
int main() {
    int values[10];
    int sum = 0;
    float mean;
    printf ("Enter 10 integer values:\n");
    for (int i=0; i<10; i++) {
        printf ("Value %d:", i+1);
        scanf ("%d", &values[i]);
        sum += values[i];
    }
    mean = (float) sum / 10;
    printf ("Sum of the 10 values: %d\n", sum);
    printf ("Mean of the 10 values: %.2f\n", mean);
    return 0;
}
```

Output: Enter 10 integer values:

```
value 1: 1
value 2: 2
value 3: 3
value 4: 4
value 5: 5
value 6: 6
value 7: 7
value 8: 8
value 9: 9
value 10: 0
```

Sum of the 10 values: 45

Mean of the 10 values: 4.5

15. Calculate sum and mean of n values.

Code: #include <stdio.h>

```
int main() {
    int n, i;
    float num, sum = 0.0, mean;
    printf ("Enter the number of values (n):");
    scanf ("%d", &n);
    printf ("Enter %d numbers:\n", n);
    for (i=0; i<n; i++) {
        printf ("Enter number %d:", i+1);
        scanf ("%f", &num);
        sum += num;
    }
    mean = sum/n;
    printf ("Sum of the numbers = %.2f\n", sum);
    printf ("Mean of the numbers = %.2f\n", mean);
    return 0;
}
```

Output: Enter the number of values (n): 2

```
Enter number 1: 5
Enter number 2: 7
Sum of the numbers = 12.00
Mean of the numbers = 6.00
```

16. Find the largest and smallest among 100 numbers.

Code: #include <stdio.h>

```
#define MAX_NUMBERS 100
int main() {
    int numbers[MAX_NUMBERS];
    int largest, smallest;
    int i;
    printf ("Enter %d numbers:\n", MAX_NUMBERS);
    for (i=0; i<MAX_NUMBERS; i++) {
        printf ("Enter number %d:", i+1);
        scanf ("%d", &numbers[i]);
    }
    largest = numbers[0];
    smallest = numbers[0];
    for (i=1; i<MAX_NUMBERS; i++) {
        if (numbers[i] > largest) {
            largest = numbers[i];
        }
        if (numbers[i] < smallest) {
            smallest = numbers[i];
        }
    }
    printf ("\nlargest number: %d\n", largest);
    printf ("smallest number: %d\n", smallest);
    return 0;
}
```

17. Count positive, negative and zero numbers among 200 values.

```
Code: #include <stdio.h>
int main() {
    int numbers[200];
    int positive_count = 0;
    int negative_count = 0;
    int zero_count = 0;
    int i;
    printf("Enter 200 integer values:\n");
    for (i=0; i<200; i++) {
        printf("Enter number %d:", i+1);
        scanf("%d", &numbers[i]);
    }
    for (i=0; i<200; i++) {
        if (numbers[i]>0) {
            positive_count++;
        }
        else if (numbers[i]<0) {
            negative_count++;
        }
        else {
            zero_count++;
        }
    }
    printf("\nResults:\n");
    printf("Positive numbers: %d\n", positive_count);
    printf("Negative numbers: %d\n", negative_count);
    printf("Zero numbers: %d\n", zero_count);
    return 0;
}
```

18. Count number of boys and girls in a class of 50 students using sex code.

Code:

```
#include <stdio.h>
#define TOTAL_STUDENTS 50
int main() {
    int boys = 0;
    int girls = 0;
    char sex_code;
    printf ("Enter 'M' for male and 'F' for female for %d students:\n", TOTAL_STUDENTS);
    for (int i=0; i < TOTAL_STUDENTS; i++) {
        printf ("Enter sex code for student %d : ", i+1);
        scanf ("%c", &sex_code);
        if (sex_code >= 'a' && sex_code <= 'z') {
            sex_code = sex_code - 32;
        }
        if (sex_code == 'M') {
            boys++;
        } else if (sex_code == 'F') {
            girls++;
        } else {
            printf ("Invalid input. Please enter 'M' or 'F'. Re-enter
                    for student %d.\n", i+1);
            i--;
        }
    }
    printf ("\nNumber of boys : %d\n", boys);
    printf ("Number of girls : %d\n", girls);
    return 0;
}
```

19. Print numbers from 1 to 100 divisible by 5.

Code: #include <stdio.h>

```
int main() {
    int i;
    printf ("Numbers from 1 to 100 divisible by 5 are:\n");
    for (i=1; i<=100; i++) {
        if (i%5 == 0) {
            printf ("%d\n", i);
        }
    }
    return 0;
}
```

Output: Numbers from 1 to 100 divisible by 5 are:

5	85
10	90
15	95
20	100
25	
30	
35	
40	
45	
50	
55	
60	
65	
70	
75	
80	

20. Print sum of all numbers from 1 to 200 divisible by 3.

Code: #include <stdio.h>

```
int main () {  
    int sum = 0;  
    for (int i = 1; i <= 200; i++) {  
        if (i % 3 == 0) {  
            sum += i;  
        }  
    }
```

```
    printf ("The sum of all numbers from 1 to 200 devisable by  
            3 is: %d\n", sum);
```

```
    return 0;  
}
```

Output: The sum of all numbers from 1 to 200 divisible by
3 is: 1683-

22. Separate digits of a given number.

Code: #include <stdio.h>

```
int main() {
    int number, digit;
    printf ("Enter a number : ");
    scanf ("%d", &number);
    if (number < 0) {
        number = -number;
    }
    if (number == 0) {
        printf ("0\n");
        return 0;
    }
    printf ("Separated digits (reverse order) : ");
    int temp_number = number;
    while (temp_number > 0) {
        digit = temp_number % 10;
        printf ("%d ", digit);
        temp_number /= 10;
    }
    printf ("\n");
    int digits[10];
    int count = 0;
    temp_number = number;
    while (temp_number > 0) {
        digits[count] = temp_number % 10;
        temp_number /= 10;
        count++;
    }
    printf ("Separated digits (original order) : ");
    for (int i = count - 1; i >= 0; i--) {
        printf ("%d ", digits[i]);
        if (i > 0)
            printf ",";
    }
    printf ("\n");
    return 0;
}
```

Output : Enter a number : 45
Separated digits (reverse order) : 5,4
Separated digits (original order) : 4,5

22. Count digits in a given number.

Code:- #include <stdio.h>

```
int main() {
    int number;
    int count = 0;
    printf ("Enter an integer:");
    scanf ("%d", &number);
    if (number == 0) {
        count = 1;
    } else {
        if (number < 0) {
            number = -number;
        }
        while (number > 0) {
            number /= 10;
            count++;
        }
    }
    printf ("Number of digits: %d\n", count);
    return 0;
}
```

Output: Enter an integer: 7

Number of digits: 1

23. Sum of digits of a given number.

Code: #include <stdio.h>

```
int main() {
    int number, temp_number, sum=0, remainders;
    printf ("Enter an integer:");
    scanf ("%d", &number);
    temp_number = number;
    while (temp_number != 0) {
        remainder = temp_number % 10;
        sum += remainder;
        temp_number /= 10;
    }
    printf ("Sum of digits of %d = %d\n", number, sum);
    return 0;
}
```

Output: Enter an integer: 7

Sum of digits of 7 = 7.

24. Reverse the digits of a number.

Code : #include <stdio.h>

```
int main() {
    int number, reversedNumber = 0; remainder;
    printf ("Enter an integer : ");
    scanf ("%d", &number);
    while (number != 0) {
        remainder = number % 10;
        reversedNumber = reversedNumber * 10 + remainder;
        number /= 10;
    }
    printf ("Reversed number = %d \n", reversedNumber);
    return 0;
}
```

Output : Enter an integer : 45

Reversed number = 54

25. Check whether a number is palindrome.

Code: #include <stdio.h>

```
int main() {
    int n, reversedN = 0, remainder, originalN;
    printf ("Enter an integer : ");
    scanf ("%d", &n);
    originalN = n;
    while (n != 0) {
        remainder = n % 10;
        reversedN = reversedN * 10 + remainder;
        n /= 10;
    }
    if (originalN == reversedN)
        printf ("%d is a palindrome.\n", originalN);
    else
        printf ("%d is not a palindrome.\n", originalN);
    return 0;
}
```

Output: Enter an integer 17

17 is a palindrome.

26. Check whether a number is an Armstrong number.

Code: #include <stdio.h>
#include <math.h>
int main() {
 int num, originalNum, remainders, m=0;
 double result = 0.0;
 printf ("Enter an integer: ");
 scanf ("%d", &num);
 for (originalNum = num; originalNum != 0; ++m) {
 originalNum /= 10;
 }
 originalNum = num;
 while (originalNum != 0) {
 remainder = originalNum % 10;
 result += pow (remainder, m);
 originalNum /= 10;
 }
 if ((int) result == num) {
 printf ("%d is an Armstrong number.\n", num);
 } else {
 printf ("%d is not an Armstrong number.\n", num);
 }
 return 0;
}

Output: Enter an integer: 77
77 is not an Armstrong number.

27. Print all factors of a number.

Code: #include <stdio.h>
int main() {
 int num, i;
 printf ("Enter a positive integer: ");
 scanf ("%d", &num);
 printf ("Factors of %d are: ", num);
 for (i=1; i<=num; ++i) {
 if (num % i == 0) {
 printf ("%d", i);
 }
 }
 printf ("\n");
 return 0;
}

Output: Enter a positive integer: 77
Factors of 77 are: 1 7 11 77

28. Check whether a number is a perfect number.

Code: #include <stdio.h>

```
int main() {
    int num, i, sum_of_divisors = 0;
    printf ("Enter a positive integer:");
    scanf ("%d", &num);
    if (num <= 0) {
        printf ("Please enter a positive integer.\n");
        return 1;
    }
    for (i = 1; i < num; i++) {
        if (num % i == 0) {
            sum_of_divisors += i;
        }
    }
    if (sum_of_divisors == num) {
        printf ("%d is a perfect number.\n", num);
    } else {
        printf ("%d is not a perfect number.\n", num);
    }
    return 0;
}
```

Output : Enter a positive integer: 6
6 is a perfect number.

* is a palindrome.

29. Check whether a number is a prime number.

Code : #include <stdio.h>

#include <math.h>

int main() {

 int n, i, flag = 0;

 printf("Enter a positive integer: ");

 scanf("%d", &n);

 if (n == 0 || n == 1) {

 flag = 1;

 } else {

 for (i = 2; i <= sqrt(n); ++i) {

 if (n % i == 0) {

 flag = 1;

 break;

 }

 }

 if (flag == 0) {

 printf("%d is a prime number.\n", n);

 } else {

 printf("%d is not a prime number.\n", n);

 }

 return 0;

}

Output: Enter a positive integer: 6

6 is not a prime number.

30. Print all prime numbers between 1 and 500.

Code:

```
#include <stdio.h>
#include <math.h>
int isPrime (int num) {
    if (num <= 1) {
        return 0;
    }
    for (int i=2; i<=sqrt(num); i++) {
        if (num % i == 0) {
            return 0;
        }
    }
    return 1;
}
int main() {
    printf ("Prime numbers between 1 and 500 are:\n");
    for (int i=2; i<=500; i++) {
        if (isPrime (i)) {
            printf ("%d ", i);
        }
    }
    printf ("\n");
    return 0;
}
```

Output: Prime numbers between 1 and 500 are:

2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79
83 89 97 101 103 107 109 113 127 131 137 139 149 151 157 163 167
173 179 181 191 193 197 199 211 223 227 229 233 239 241 251 257
263 269 271 277 281 283 293 307 311 313 317 331 337 347 349 353
359 367 373 379 383 389 397 401 409 419 421 431 433 439 443 449
457 461 463 467 479 487 491 499.

Q: 31. Find summation of prime numbers between 1 to 500.
 (Code: #include <stdio.h>
 #include <math.h>
 int isPrime (int num) {
 if (num <= 1) {
 return 0;
 }
 for (int i=2; i<=sqrt(num); i++) {
 if (num % i == 0) {
 return 0;
 }
 }
 return 1;
 }
 int main() {
 int sum = 0;
 printf ("prime numbers between 1 to 500 are :\n");
 for (int i=2; i<=500; i++) {
 if (isPrime (i)) {
 printf ("%d", i);
 sum += i;
 }
 }
 printf ("\n\nSummation of prime numbers between 1 and 500 is : %d\n", sum);
 return 0;
 }

Output: Prime numbers between 1 and 500 are:

2 3 5 7 11 467 479 487 491 499.

Summation of prime numbers between 1 and 500 is: 21536.

32. Count how many prime numbers are there between 1 and 500.

Code: #include <stdio.h>

```
int isPrime (int num) {
    if (num <= 1) {
        return 0;
    }
    for (int i = 2; i * i <= num; i++) {
        if (num % i == 0) {
            return 0;
        }
    }
    return 1;
}

int main() {
    int count = 0;
    printf ("Prime numbers between 1 and 500 are:\n");
    for (int i = 1; i <= 500; i++) {
        if (isPrime(i)) {
            printf ("%d ", i);
            count++;
        }
    }
    printf ("\nTotal prime numbers between 1 and 500 : %d\n", count);
    return 0;
}
```

Output: Total prime numbers between 1 and 500 : 95.

33. Check whether a number is an automorphic number. ($25 \rightarrow 625$)

Code :

```
#include <stdio.h>
#include <math.h>
int main() {
    int n, temp, square_n, last_digits;
    int count = 0;
    int power_of_10 = 1;
    printf ("Enter a number:");
    scanf ("%d", &n);
    square_n = n * n;
    temp = n;
    while (temp > 0) {
        temp /= 10;
        count++;
    }
    for (int i = 0; i < count; i++) {
        power_of_10 *= 10;
    }
    last_digits = square_n % power_of_10;
    if (last_digits == n) {
        printf ("%d is an automorphic number.\n", n);
    } else {
        printf ("%d is not an automorphic number.\n", n);
    }
    return 0;
}
```

Output: Enter a number: 625
625 is an automorphic number.

34. Print Fibonacci series up to n terms.

Code: #include <stdio.h>

```
int main() {
    int n, t1 = 0, t2 = 1, nextTerm;
    printf ("Enter the number of terms : ");
    scanf ("%d", &n);
    if (n == 1) {
        printf ("Fibonacci Series : %d\n", t1);
    } else if (n == 2) {
        printf ("Fibonacci Series : %d, %d\n", t1, t2);
    } else {
        printf ("Fibonacci Series : %d, %d", t1, t2);
        for (int i = 3; i <= n; ++i) {
            nextTerm = t1 + t2;
            printf (" , %d", nextTerm);
            t1 = t2;
            t2 = nextTerm;
        }
        printf ("\n");
    }
    return 0;
}
```

Output: Enter the number of terms : 7
Fibonacci Series : 0, 1, 1, 2, 3, 5, 8.

Section C : n

Pattern Set 1

A] 11
12
13
21
22
23

Code: #include <stdio.h>
int main() {
 int i, j;
 for (i = 1; i <= 2; i++) {
 for (j = 1; j <= 2; j++) {
 printf ("%d %d\n", i, j);
 }
 }
 return 0;
}

B] 11
21
22
31
..
32
33 ..

Code: #include <stdio.h>
int main() {
 for (int i=1; i<=3; i++) {
 for (int j=1; j<=i; j++) {
 printf ("%d %d\n", i, j);
 }
 }
 return 0;
}

c) 111
112
121
122
211
212
221
222

Code : #include <stdio.h>

```
int main() {
    int i, j, k;
    for (i = 1; i <= 2; i++) {
        for (j = 1; j <= 2; j++) {
            for (k = 1; k <= 2; k++) {
                printf ("%d%d%d\n", i, j, k);
            }
        }
    }
    return 0;
}
```

D] 5 1
5 2
4 1
4 2
3 1
3 2

Code: #include <stdio.h>
int main() {
 for (int i=5; i>=3; i--) {
 for (int j=1; j<=2; j++) {
 printf("%d%d\n", i, j);
 }
 }
 return 0;

Pattern Set 2

A) 15

24

33

42

51

```
Code: #include<stdio.h>
int main() {
    int i, j;
    for (i = 1; i <= 5; i++) {
        j = 6 - i;
        printf ("%d %d\n", i, j);
    }
    return 0;
}
```

B) 15
15
24
24
33
33

Code: #include <stdio.h>
int main() {
 int i;
 for (i=1; i<=3; i++) {
 printf ("%d %d \n", i, 6-i);
 printf ("%d %d \n", i, 6-i);
 }
 return 0;
}