```java
class Test1
{



class Test1
{


        public static void main(String[] args)
        {
                int day = 11;

                switch (day)
                {

                        case 1:
                                System.out.println("Sunday");
                                break;

                        case 2:
                                System.out.println("Monday");
                                break;

                        case 3:
                                System.out.println("Tuesday");
                                break;

                        case 4:
                                System.out.println("Wednesday");
                                break;

                        case 5:
                                System.out.println("Thursday");
                                break;
                        case 6:
                                System.out.println("Friday");
                                break;
                        case 7:
                                System.out.println("saturday");
                                break;

                        default :
                        {
                                System.out.println("enter a valid number");
                        }
                }
```

```
        }
}
```

LOOPING STATEMENTS

- IN PROGRAMMING LANGUAGE LOOPS ARE USED TO EXECUTE THE SET OF INSTRUCTIONS/FUNCTIONS REPEATEDLY WHEN SOME GIVEN CONDITION BECOMES TRUE
- THERE ARE THREE TYPES OF LOOPS IN JAVA
  - FOR LOOP
  - WHILE LOOP
  - DO-WHILE LOOP

FOR LOOP

SYNTAX :

```
For(intialise; expression/condition; increment/decrement)
{

        //code to be executed
}
```

- **INITIALIZE :**

  TO INITIALISE THE VALUE TO THE VARIABLE. IN FOR LOOP THIS STATEMENT EXECUTES ONLY ONCE

- **EXPRESSION/CONDITION**

  IT IS TO GIVE THE CONDITION FOR HOW MANY TIMES THE LOOP SHOULD BE REPATED. IF THE CONDITION IS TRUE THEN IT ENTERS THE LOOP ELSE IT EXITS THE LOOP.

- **INCREMENT/DECREMENT**

  IT IS USED TO INCREMENT /DECREMENT THE INTIALIZED VALUE SO THAT THE LOOP DOESN'T ENTER THE INFINITE STAGE

```
                1       2,5,8    4,7,10
        for (int i=1; i<=100; i++)
        {
                3,6,9
                System.out.println(i);
        }
```

WHILE LOOP

- THE WHILE LOOP IS USED TO ITERATE A PART OF THE PROGRAM SEVERAL TIMES
- IF THE NUMBER OF ITERATION IS NOT FIXED , IT IS RECOMMENDED TO USE WHILE LOOP

SYNTAX :

```
WHILE(CONDITION)
{
        //CODE TO BE EXECUTED
        INCREMENT/DECREMENT
}
```

- WAP TO PRINT FIRST 10 NATURAL NUMBERS

```
class Test1
{

        public static void main(String[] args)
        {

                int i = 1;

                while (i<=10)
                {
                        System.out.println(i);
                        i++;
                }

        }
}
```

DO-WHILE LOOP

- THE JAVA DO-WHILE LOOP IS USED TO ITERTATE A PART OF THE PROGRAM SEVERAL TIMES
- IF THE NUMBER OF ITERATION IS NOT FIXED AND YOU MUST HAVE TO EXECUTE THE LOOP AT LEAST ONCE, IT IS RECOMMENDED TO USE DO-WHILE LOOP
- THE JAVA DO-WHILE LOOP IS EXECUTED AT LEAST ONCE BECAUSE CONDITION IS CEHCKED AFTER THE LOOP BODY

SYNTAX :

```
Do
{

        //code to be executed
        Increment/decrement
}while(condition);
```

WHILE	DO WHILE
CONDITION CHECK HAPPENS IN THE BEGINNING	CONDITION CHECK HAPPENS AT THE END OF THE LOOP
IN GENERAL THE STATEMENT ARE NOT EXECUTED UNTIL CONDITION IS CHECKED AND IT BECOMES TRUE	IN GENERAL THE STATEMENT ARE EXECUTES AT LEAST ONCE EVEN THOUGH THE CONDITION IS FALSE

Example :

```java
class Test1
{

    public static void main(String[] args)
    {

        int i = 1;
        int num = 10;

        do
        {
            if (i%2 == 0)
            {
                System.out.println(i + " is even");
            }
            else
            {
                System.out.println(i + " is odd");
            }
            i++;
        }
        while (i<=num);
    }
}
```

WAP TO CHECK THE GIVEN NUMBER IS PALINDROME OR NOT

Int num = 123

321 ----> 3

Rev = 0

```
        do
        {
            r = num%10;
            rev = (rev*10) + r;
            num = num/10;--> 0
```

r=num%10	Num=num/10	Rev=(rev*10)+r
3	12	3
2	1	32
1	0	321

```
}
while (num>0);
s.o.p(rev)
```

0