Programming Using C
Week 03-1

Decision Making and Branching - if...else if and switch…case

Prepared By: REC Faculty 4.0 Team

Selection structure:
if-else if ladder

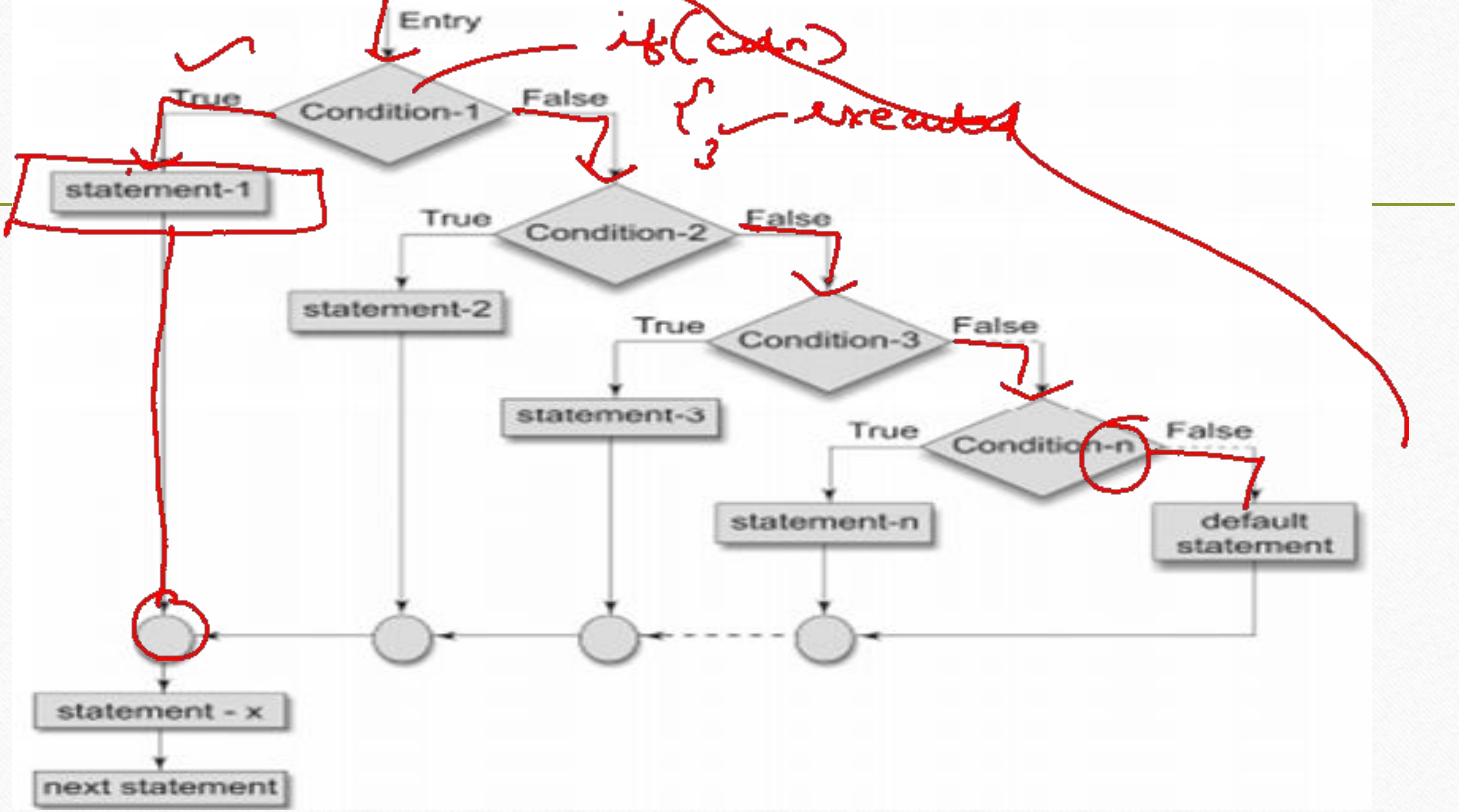
- if-else-if ladder helps user to **decide among multiple conditions.**
- **Multiple conditions are checked to make a single decision.**

Syntax of if...else Ladder

```
if (test expression1) {
   // statement(s)
}
else if(test expression2) {
   // statement(s)
}
else if (test expression3) {
   // statement(s)
}
.
.
else {
   // statement(s)
}
```

- if statements are executed from the top down.
- As soon as one of the conditions controlling the if is true, the statement associated with that if is executed, and the rest of the else-if ladder is bypassed.
- If none of the conditions is true, then the final else statement will be executed.

FLOWCHART


Entry
True
Condition-1
False
statement-1
True
Condition-2
False
statement-2
True
Condition-3
False
statement-3
True
Condition-n
False
statement-n
default statement
statement - x
next statement
if(cdn)
{ — executed
}

Write a program in C to calculate and print the Electricity bill where the unit consumed by the user is given from test case. It prints the total amount the customer has to pay. The charges are as follows:

Unit Charge / Unit

Upto 199 @1.20

200 and above but less than 400 @1.50

400 and above but less than 600 @1.80

600 and above @2.00

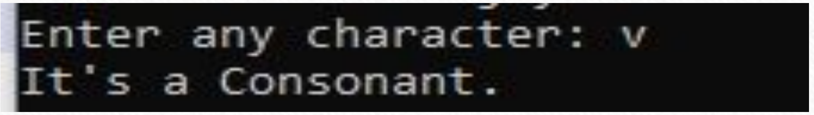
If bill exceeds Rs.400 then a surcharge of 15% will be charged and the minimum bill should be of Rs.100/-

```
#include<stdio.h>
#include<math.h>
int main()
{
    int unit;
    float bill;
    scanf("%d", &unit);
    if(unit<=199)
              bill=1.20*unit;
        else if(unit>=200 && unit<400)
            bill=1.50*unit;
    else if(unit>=400 && unit<600)
            bill=1.80*unit;

else
        bill=2.00*unit
```

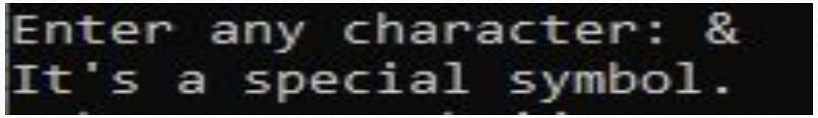
C program to read a character and print it’s a vowel / consonant / number / special digit.

```
#include <stdio.h>
int main()
{
    char ch;

    printf("Enter any character: ");
    scanf("%c", &ch);
```

```
if(ch=='a' || ch=='e' || ch=='i' || ch=='o' ||
     ch=='u' || ch=='A' || ch=='E' || ch=='I' ||
      ch=='O' || ch=='U')
      {
      printf("It's Vowel.");
      }
  else if((ch >= 'a' && ch <= 'z') || (ch >= 'A'
                                          && ch <= 'Z'))
  {
    printf("It's a Consonant.");
  }


  else if (ch>='0' && ch<='9')
     {
            printf("It's a number.");
```



```
Enter any character: v
It's a Consonant.
```



```
Enter any character: &
It's a special symbol.
```

Write a program to determine the type of berth when the seat/berth number in the train is given.

Input Format:

Input consists of a single integer.
Assume that the range of input is between 1 and 72.

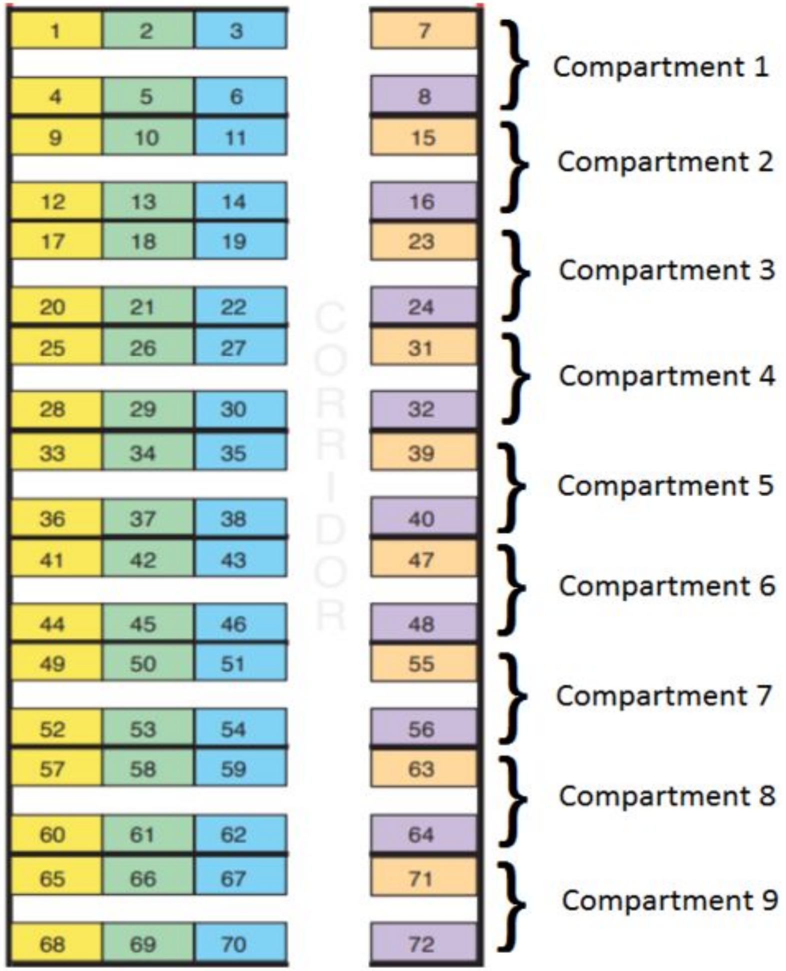
Output Format:

Output consists of a single string.
[Upper or Middle or Lower or Side Lower or Side Upper]

For example:

Input
9

Result
Lower Berth

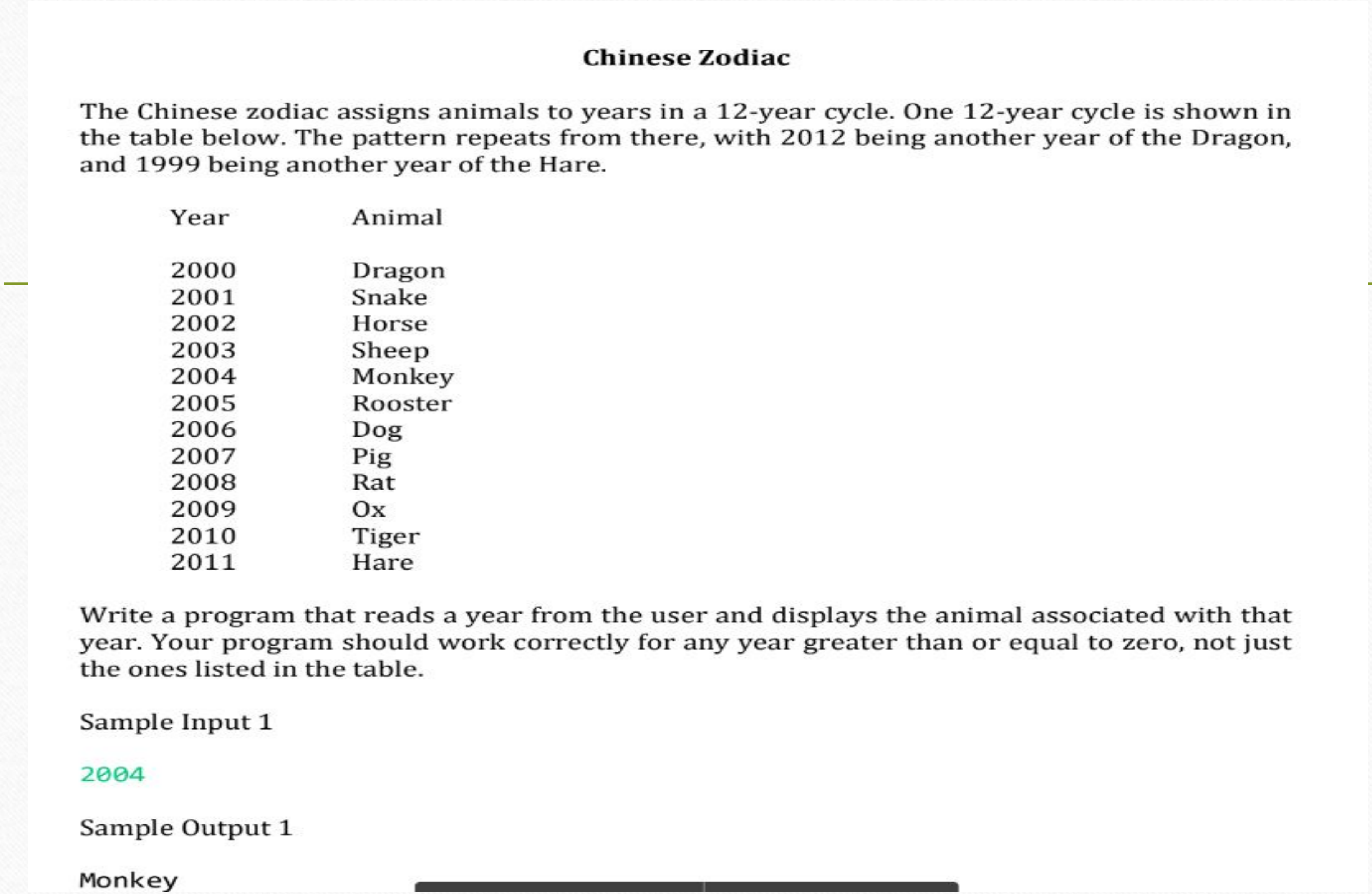


1	2	3	7	Compartment 1
4	5	6	8	
9	10	11	15	Compartment 2
12	13	14	16	
17	18	19	23	Compartment 3
20	21	22	24	
25	26	27	31	Compartment 4
28	29	30	32	
33	34	35	39	Compartment 5
36	37	38	40	
41	42	43	47	Compartment 6
44	45	46	48	
49	50	51	55	Compartment 7
52	53	54	56	
57	58	59	63	Compartment 8
60	61	62	64	
65	66	67	71	Compartment 9
68	69	70	72	

🟨	Lower berth
🟩	Middle berth
🟦	Upper berth
🟧	Side Lower berth
🟪	Side Upper berth

Color codes of seats

```
/* Train seat checking */

#include<stdio.h>
int main()
{
    int n;
    scanf("%d", &n);

    if(n%8==1||n%8==4)
             printf("Lower Berth");


    else if(n%8==2||n%8==5)
            printf("Middle Berth");
```



Chinese Zodiac

The Chinese zodiac assigns animals to years in a 12-year cycle. One 12-year cycle is shown in the table below. The pattern repeats from there, with 2012 being another year of the Dragon, and 1999 being another year of the Hare.

Year	Animal
2000	Dragon
2001	Snake
2002	Horse
2003	Sheep
2004	Monkey
2005	Rooster
2006	Dog
2007	Pig
2008	Rat
2009	Ox
2010	Tiger
2011	Hare

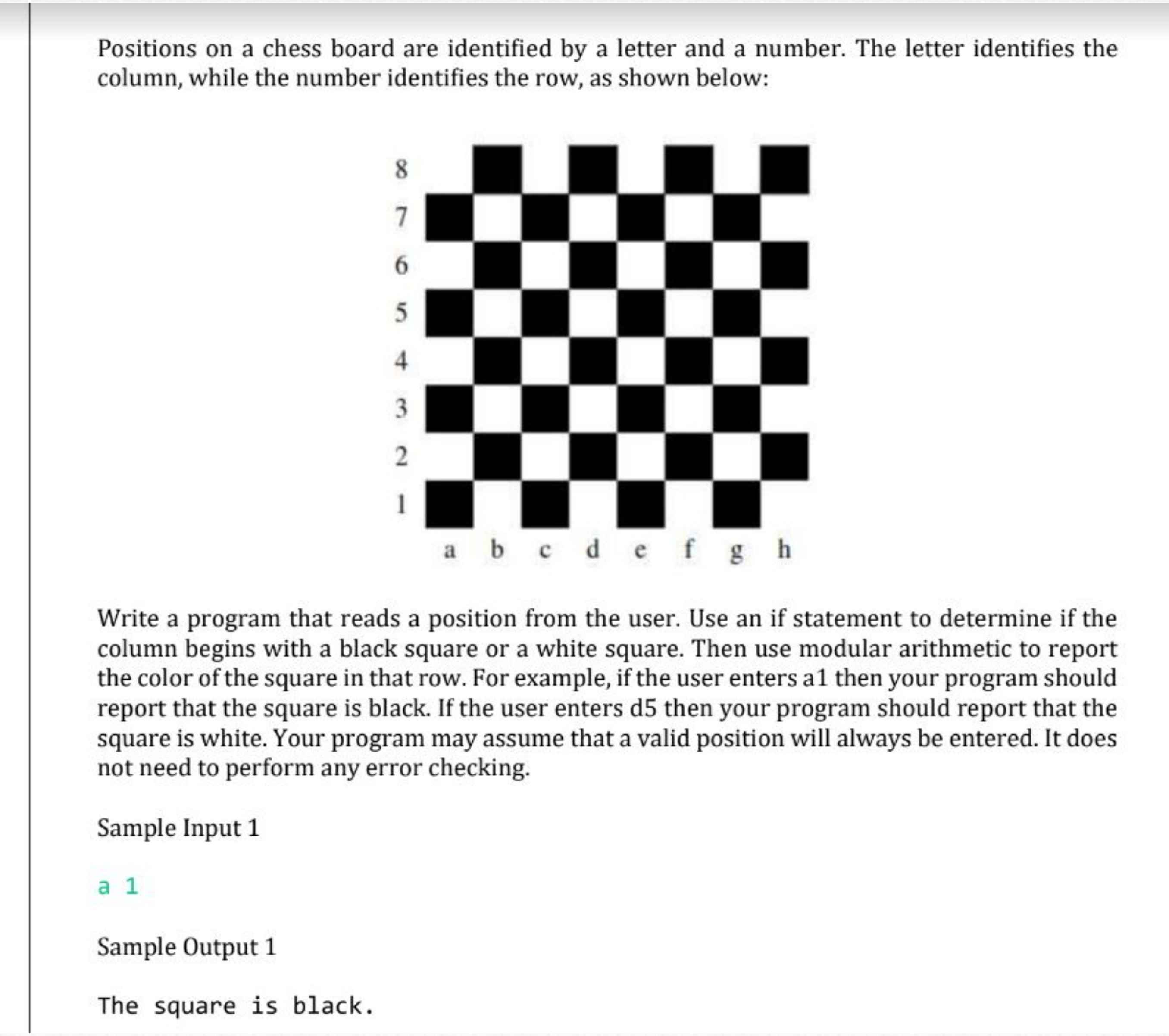
Write a program that reads a year from the user and displays the animal associated with that year. Your program should work correctly for any year greater than or equal to zero, not just the ones listed in the table.

Sample Input 1

```
2004
```

Sample Output 1

```
Monkey
```



Positions on a chess board are identified by a letter and a number. The letter identifies the column, while the number identifies the row, as shown below:

Write a program that reads a position from the user. Use an if statement to determine if the column begins with a black square or a white square. Then use modular arithmetic to report the color of the square in that row. For example, if the user enters a1 then your program should report that the square is black. If the user enters d5 then your program should report that the square is white. Your program may assume that a valid position will always be entered. It does not need to perform any error checking.

Sample Input 1

```
a 1
```

Sample Output 1

```
The square is black.
```

SWITCH CASE STATEMENT

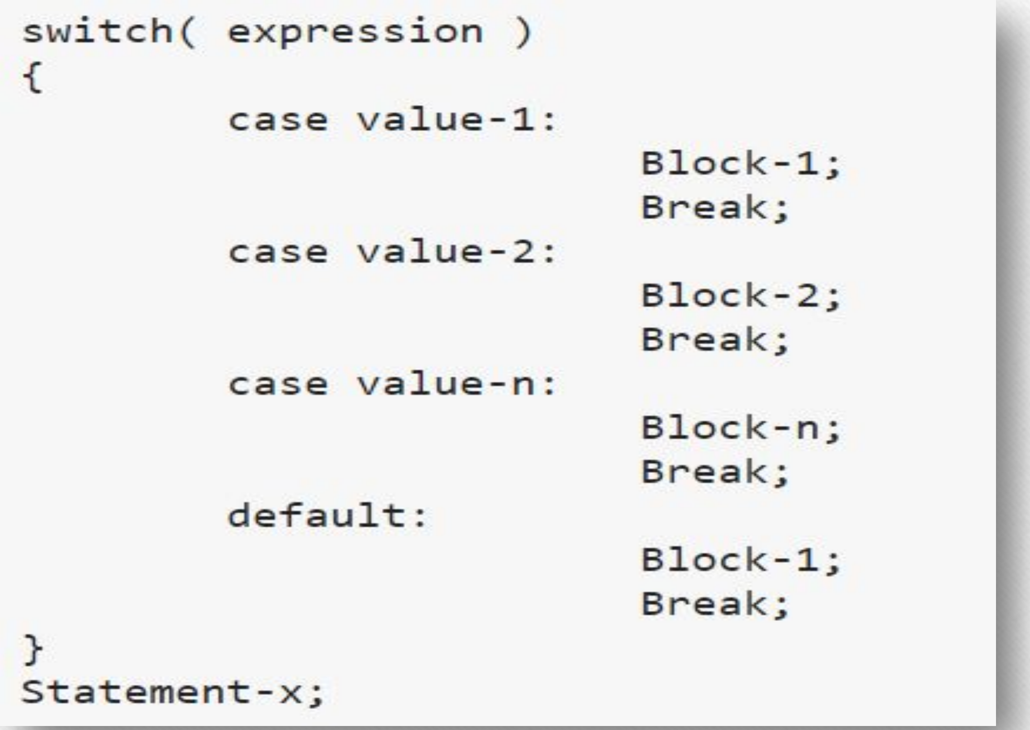
- The most flexible selection program control.
- Enables the program to execute different statements based on the result of an condition or expression ,that can have more than two values(not only true or false).
- Also called multiple choice statements.

SWITCH CASE STATEMENT

- The if statement were limited to evaluating an expression that could have only two logical values: TRUE or FALSE.
- If more than two values, have to use nested if.
- The switch statement makes such nesting unnecessary.
- Used together with case and break the program flows.

SWITCH CASE STATEMENT

The switch constructs has the following form:



```
switch( expression )
{
        case value-1:
                        Block-1;
                        Break;
        case value-2:
                        Block-2;
                        Break;
        case value-n:
                        Block-n;
                        Break;
        default:
                        Block-1;
                        Break;
}
Statement-x;
```

SWITCH CASE STATEMENT

- Evaluates the (condition/expression) and compares its value with the values following each case label.
- If a match is found between condition/expression result and one of the case values, execution is transferred to the statement(s) that follows the case label and other cases will be ignored.
- If no match is found, execution is transferred to the statement(s) following the optional default label.
- If no match is found and there is no default label, execution passes to statement-X.

SWITCH CASE STATEMENT

- To ensure that only the statements associated with the matching value are executed, include a break keyword where needed, which terminates the entire switch statement.
- The break statement is used inside the switch to terminate a statement sequence. When a break statement is reached, the switch terminates, and the flow of control jumps to the next line following the switch statement.
- The break statement is optional. If omitted, execution will continue on into the next case. The flow of control will fall through to subsequent cases until a break is reached.

SWITCH CASE EXAMPLES-general

```
#include<stdio.h>
int main()
{
    int x;
    scanf("%d",&x);
    switch(x)
    {
        case 1:
                printf("Choice 1");
                break;
        case 2:
                printf("Choice 2");
                break;
        default:
                 printf("choice other than 1 and 2");
                break;
    }
}
```

Sample Output:

```
1
Choice 1
```

SWITCH CASE EXAMPLES-default is optional

```
#include<stdio.h>
int main()
{
    int x;
    scanf("%d",&x);
    switch(x)
    {
      case 1:printf("Choice 1");
               break;
      case 2:printf("Choice 2");
                  break;
    }
}
```

Sample output

```
2
Choice 2
```

SWITCH CASE EXAMPLES-break is optional

```
#include<stdio.h>
int main()
{
    int x;
    scanf("%d",&x);
    switch(x)
    {
        case 1:printf("Choice 1");
        case 2:printf("Choice 2");
        case 3:printf("Choice 3");
               break;
        default: printf("nothing");
}
}
```

Sample output

```
2
Choice 2  Choice 3
```

SWITCH CASE -char constant allowed

```
#include<stdio.h>
int main()
{
    char x;
    scanf("%c",&x);
    switch(x)
    {
        case '1':printf("Choice 1");
                break;
        case '2':printf("Choice 2");
                break;
        case 'A':printf("Choice 3");
             break;
        default: printf("nothing");
    }
}
```

65

Error

Sample output

```
2
Choice 2
```

SWITCH CASE : integer and char constant-allowed (ASCII values compared)

```
#include<stdio.h>
int main()
{
        char x;
        scanf("%c",&x);
        switch(x)
        {             B
                case 66:
                    printf("Choice 1");
                    break
                case 67:
                    printf("Choice 2");
                    break
                case 'A':
                    printf("Choice 3");
                        break;
                default: printf("nothing");
        }
}}
```

Sample output

```
B
Choice 1
```

SWITCH CASE : (duplicate case)

```
#include<stdio.h>
int main()
{
        char x;
        scanf("%c",&x);
        switch(x)
        {
                case 65:
                    printf("Choice 1");
                    break
                case 67:
                    printf("Choice 2");
                    break
                case 'A':
                    printf("Choice 3");
                        break;
                default: printf("nothing");
        }
}}
```

output

error: duplicate case value

SWITCH CASE –(default in the front)

```
#include<stdio.h>
int main()
{
    int x;
    scanf("%d",&x);
    switch(x)
    {
        default: printf("nothing");
        case 1:printf("Choice 1");   break;
        case 2:printf("Choice 2"); break;
        case 3:printf("Choice 3"); break;
    }
}
```

Sample output1:

```
3
Choice 3
```

Sample output2:

```
4
nothingChoice 1
```

SWITCH CASE –(statement before case)

```
#include<stdio.h>
int main()
{
      int x;
      scanf("%d",&x);

      switch(x)
      {
            x=x+1;                  //will not be executed
            case 1:printf("Choice 1");

            case 2:printf("Choice 2");
            case 3:printf("Choice 3");
      }
}
```

SWITCH CASE –(statement in between cases)

```
#include<stdio.h>
int main()
{
        int x;
        scanf("%d",&x);
        switch(x)
        {
                x=x+1;                  //will not be executed
                case 1:printf("Choice 1");
                           break;
                  x=x+1;                //will not be executed
                case 2:printf("Choice 2");
                case 3:printf("Choice 3");
        }
        printf("\n%d",x);
}
```

SWITCH CASE –(templates as expression)

```
#include<stdio.h>
int main()
{
    int x;
    scanf("%d",&x);
    switch(x)
    {
        case 1+1*1:printf("Choice 2");
              break;
        case 2+1*1:printf("Choice 3");
        break;
        case 3+1*1:printf("Choice 4");
        break;
    }
}
```

SWITCH CASE –(#define usage)

```
#include<stdio.h>
#define ODD 1
#define EVEN 0
int main()
{
    int x;
    scanf("%d",&x);
    switch(x%2)
    {
        case EVEN:
                printf("odd");
                break;
        case ODD:
                printf("even");
                break;
    }
}
```

Sample output:

```
8
odd
```

SWITCH CASE variables not allowed as templates

```
#include<stdio.h>
int main()
{
    int x;
    scanf("%d",&x);
    int a=1,b=2,c=3;
    switch(x)
    {
        case a:printf("Choice 1");
        case b:printf("Choice 2");
        case c:printf("Choice 3");
    }
}
```

Output:
case label does not reduce to an integer

Program to check whether a number is even or odd using switch case

```
#include<stdio.h>
int main()
 {
int n;
scanf("%d",&n);
switch(n%2)
{
case 0:
       printf("The given is even number");
       break;
 case 1:
       printf("The given is odd number");
       break;
}
}
```

/*C program to design calculator with basic operations using switch.*/

```
#include <stdio.h>
int main()
{
   int num1,num2;
   float result;
   char ch;

   printf("Enter first number: ");
   scanf("%d",&num1);
   printf("Enter second number: ");
   scanf("%d",&num2);

   printf("Choose operation to perform (+,-,*,/,%): ");
   scanf(" %c",&ch);
```

```
 switch(ch)
   {
       case '+':
           result=num1+num2;
           break;
      case '-':
           result=num1-num2;
           break;
      case '*':
           result=num1*num2;
           break;
     case '/':
           result=num1/num2;
           break;
    case '%':
           result=num1%num2;
           break;
       default:
           printf("Invalid operation.\n");
   }
   printf("Result: %d %c %d = %f\n",num1,ch,num2,result);
   return 0;
}
```

Sample output:

```
    Enter first number: 10
    Enter second number: 20
    Choose operation to perform (+,-,*,/,%): +
    Result: 10 + 20 = 30.000000
```

Write a C program to find the number of days in a month Using Only Switch Case statement

Sample Input and output:

Test case 1:

input:

12

Output:

Number of days in this month is: 31

Test case 2:

input:

2

Output:

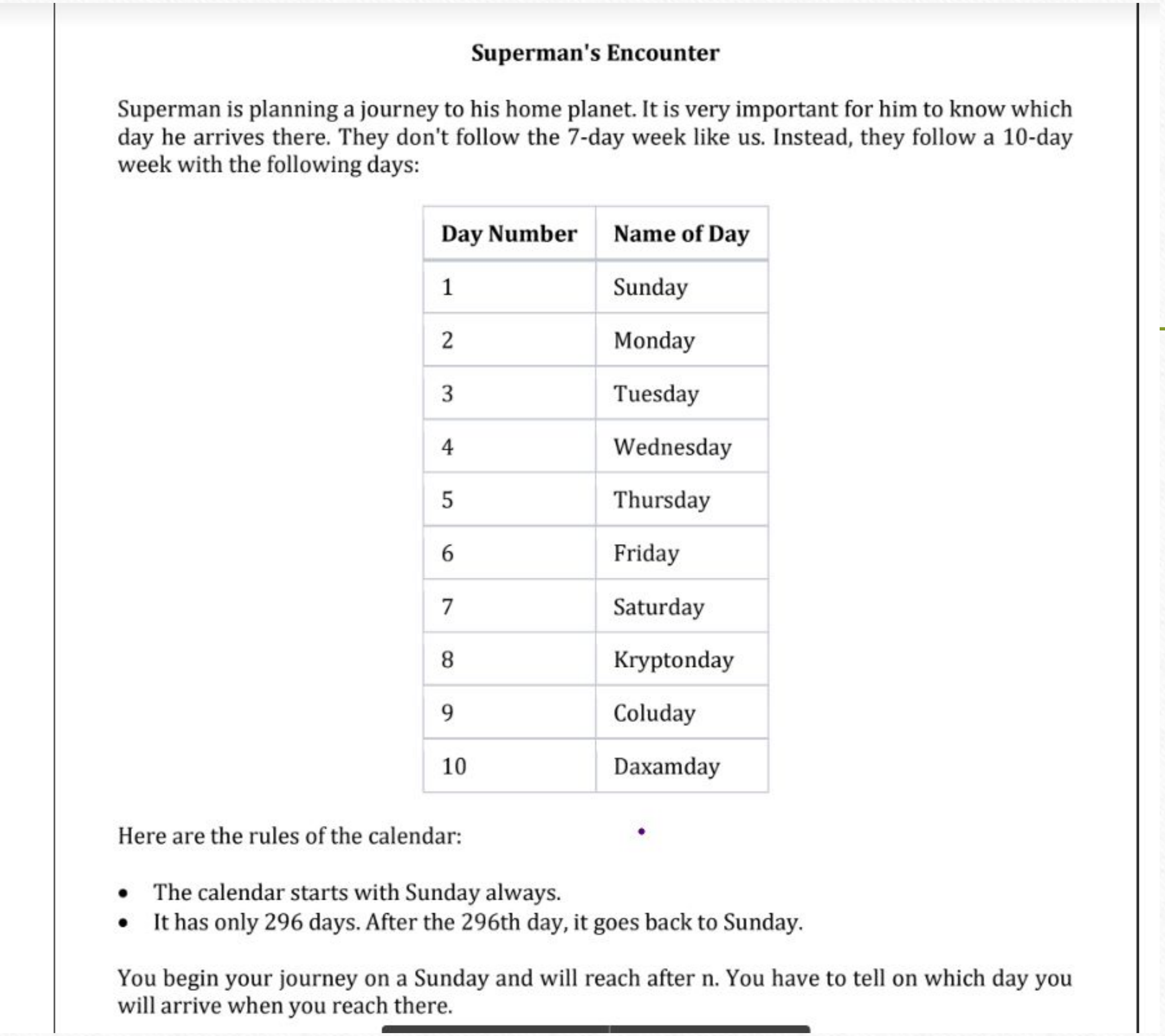
Number of days in this month is: 28

```
#include<stdio.h>
int main()
{
   int month;
   scanf("%d",&month);
   switch(month)
   {
      case 1:
      case 3:
      case 5:
      case 7:
      case 8:
      case 10:
      case 12:
            printf("Number of days in
              this month is: 31");
            break;
```

Sample output:

```
12
Number of days in this month is : 31
```

```
      case 2:
```



Superman's Encounter

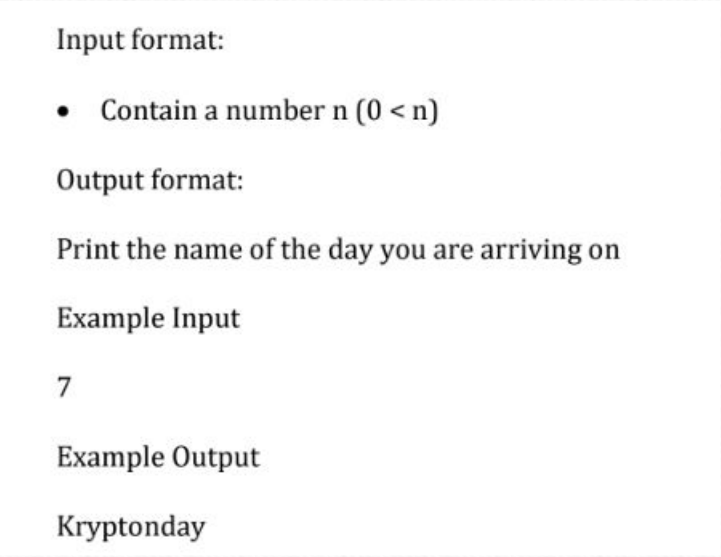
Superman is planning a journey to his home planet. It is very important for him to know which day he arrives there. They don't follow the 7-day week like us. Instead, they follow a 10-day week with the following days:

Day Number	Name of Day
1	Sunday
2	Monday
3	Tuesday
4	Wednesday
5	Thursday
6	Friday
7	Saturday
8	Kryptonday
9	Coluday
10	Daxamday

Here are the rules of the calendar:

- The calendar starts with Sunday always.
- It has only 296 days. After the 296th day, it goes back to Sunday.

You begin your journey on a Sunday and will reach after n. You have to tell on which day you will arrive when you reach there.



Input format:

- Contain a number n (0 < n)

Output format:

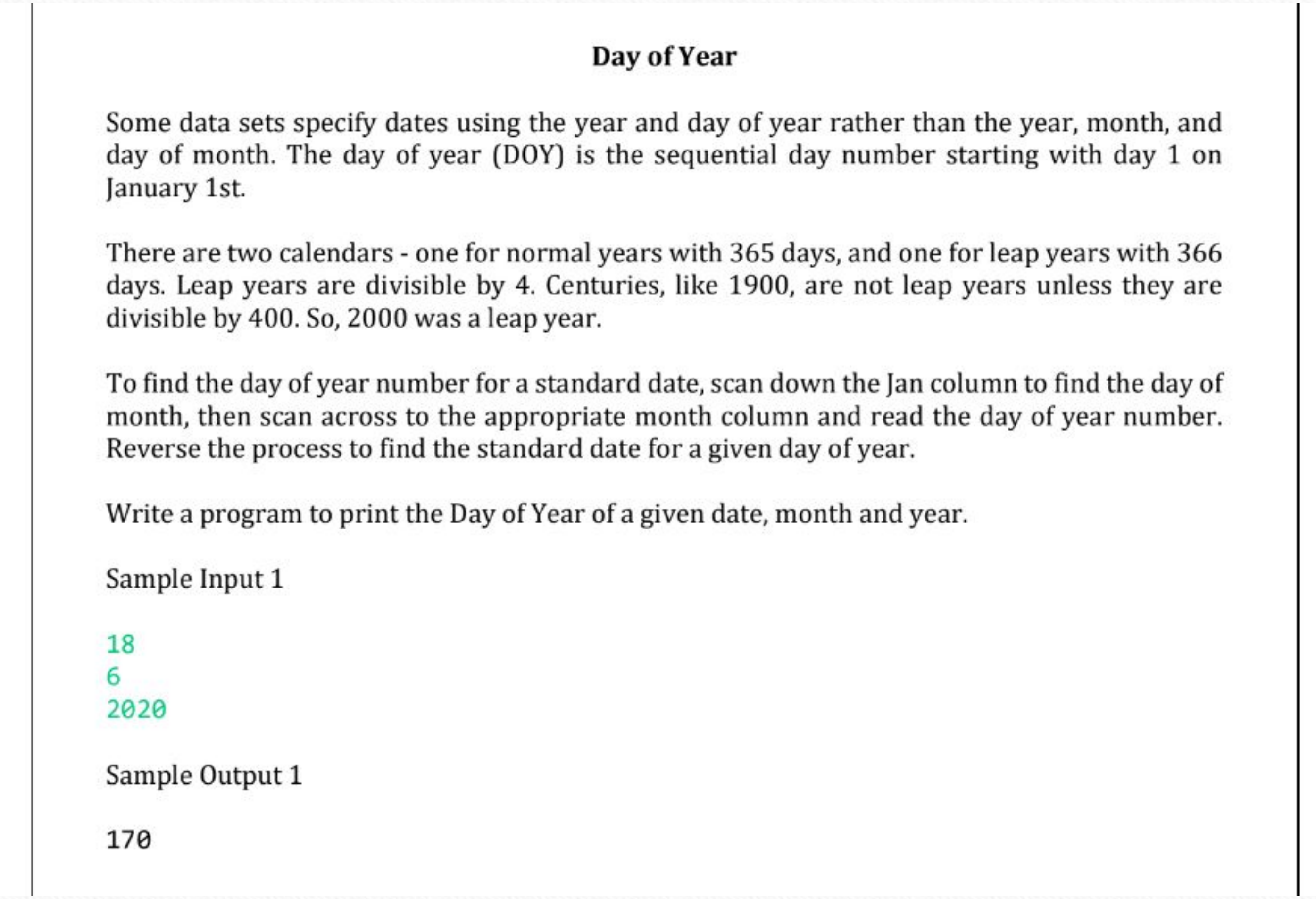
Print the name of the day you are arriving on

Example Input

7

Example Output

Kryptonday



Day of Year

Some data sets specify dates using the year and day of year rather than the year, month, and day of month. The day of year (DOY) is the sequential day number starting with day 1 on January 1st.

There are two calendars - one for normal years with 365 days, and one for leap years with 366 days. Leap years are divisible by 4. Centuries, like 1900, are not leap years unless they are divisible by 400. So, 2000 was a leap year.

To find the day of year number for a standard date, scan down the Jan column to find the day of month, then scan across to the appropriate month column and read the day of year number. Reverse the process to find the standard date for a given day of year.

Write a program to print the Day of Year of a given date, month and year.

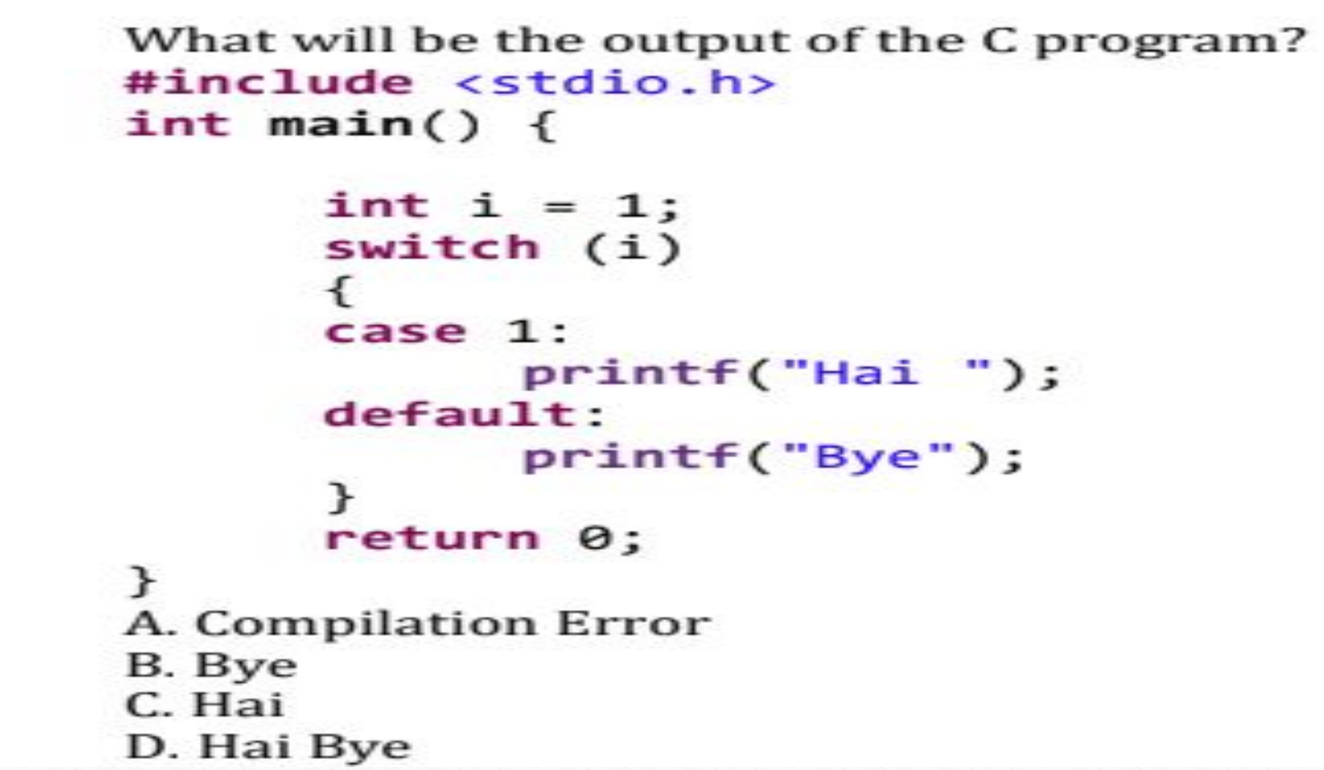
Sample Input 1

```
18
6
2020
```

Sample Output 1

```
170
```

MCQS



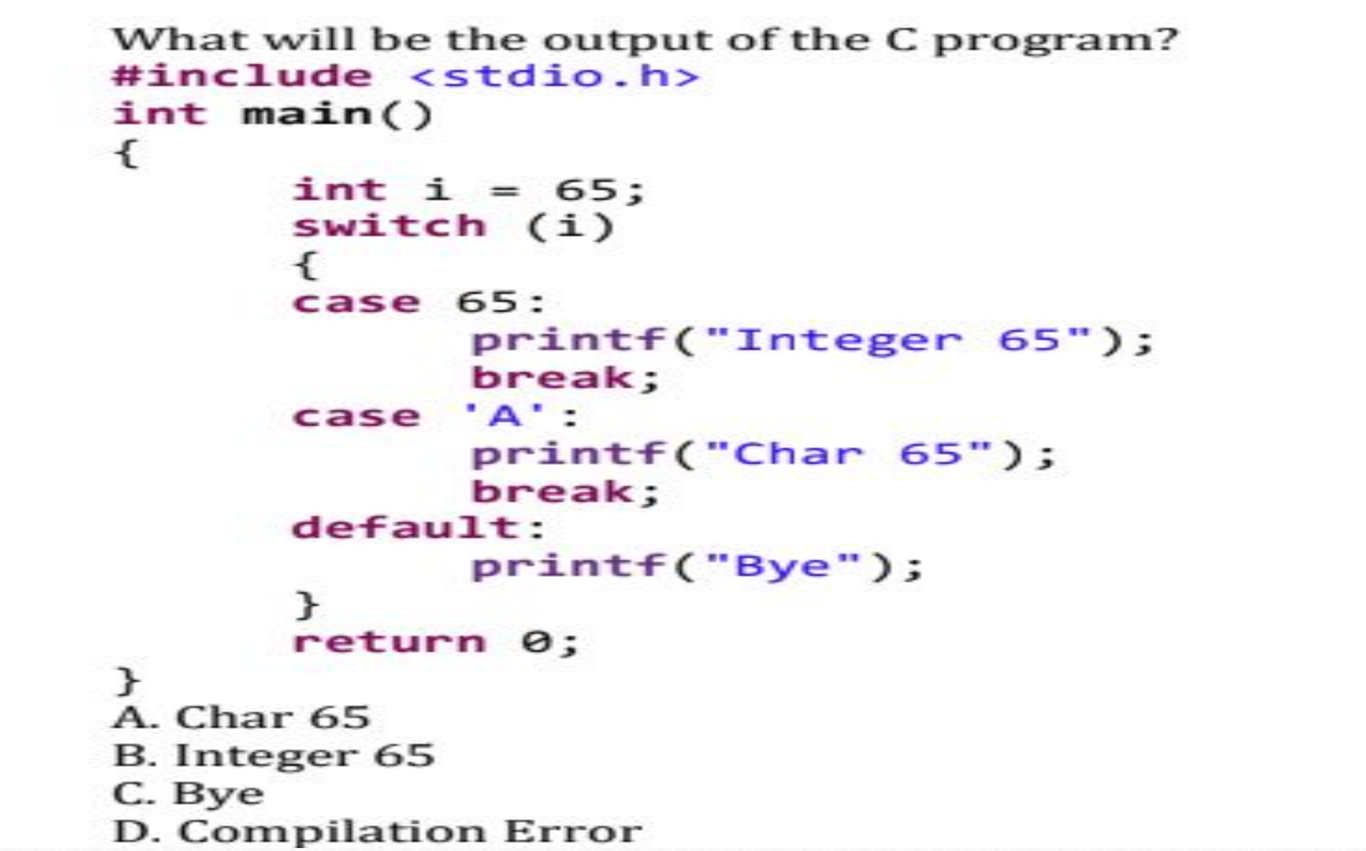
What will be the output of the C program?

```
#include <stdio.h>
int main() {

     int i = 1;
     switch (i)
     {
     case 1:
             printf("Hai ");
     default:
             printf("Bye");
     }
     return 0;
}
```

A. Compilation Error
B. Bye
C. Hai
D. Hai Bye

ANSWER:D

MCQS



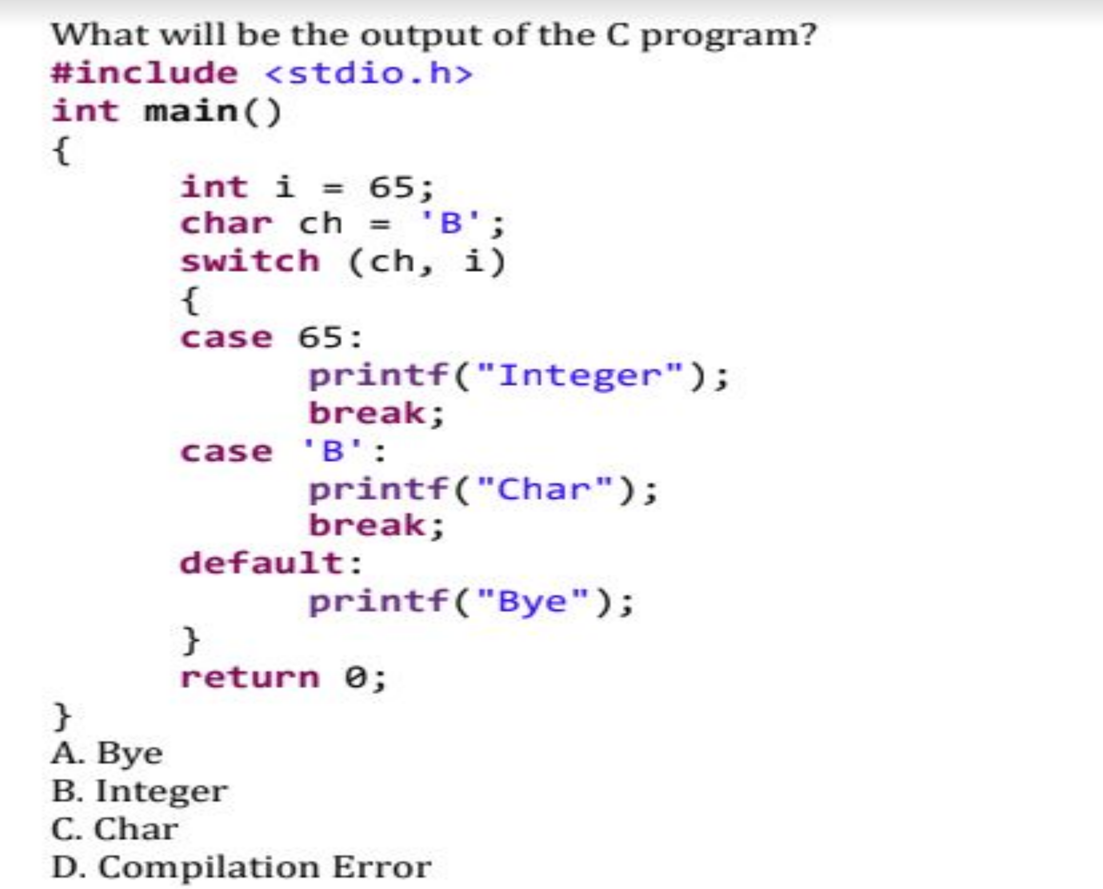
What will be the output of the C program?

```
#include <stdio.h>
int main()
{
        int i = 65;
        switch (i)
        {
        case 65:
                printf("Integer 65");
                break;
        case 'A':
                printf("Char 65");
                break;
        default:
                printf("Bye");
        }
        return 0;
}
```

A. Char 65
B. Integer 65
C. Bye
D. Compilation Error

ANSWER:D

MCQS



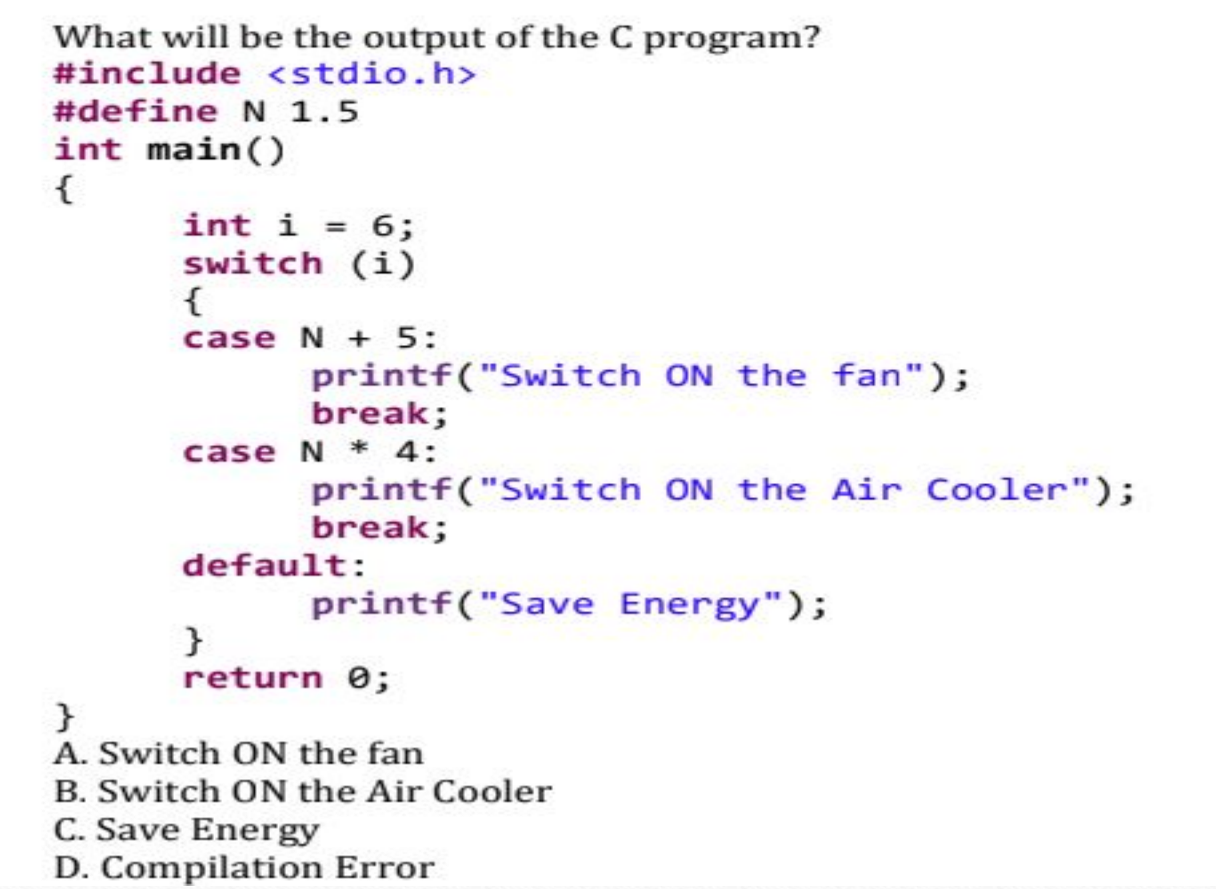
What will be the output of the C program?

```
#include <stdio.h>
int main()
{
        int i = 65;
        char ch = 'B';
        switch (ch, i)
        {
        case 65:
                printf("Integer");
                break;
        case 'B':
                printf("Char");
                break;
        default:
                printf("Bye");
        }
        return 0;
}
```

A. Bye
B. Integer
C. Char
D. Compilation Error

ANSWER:B

MCQS



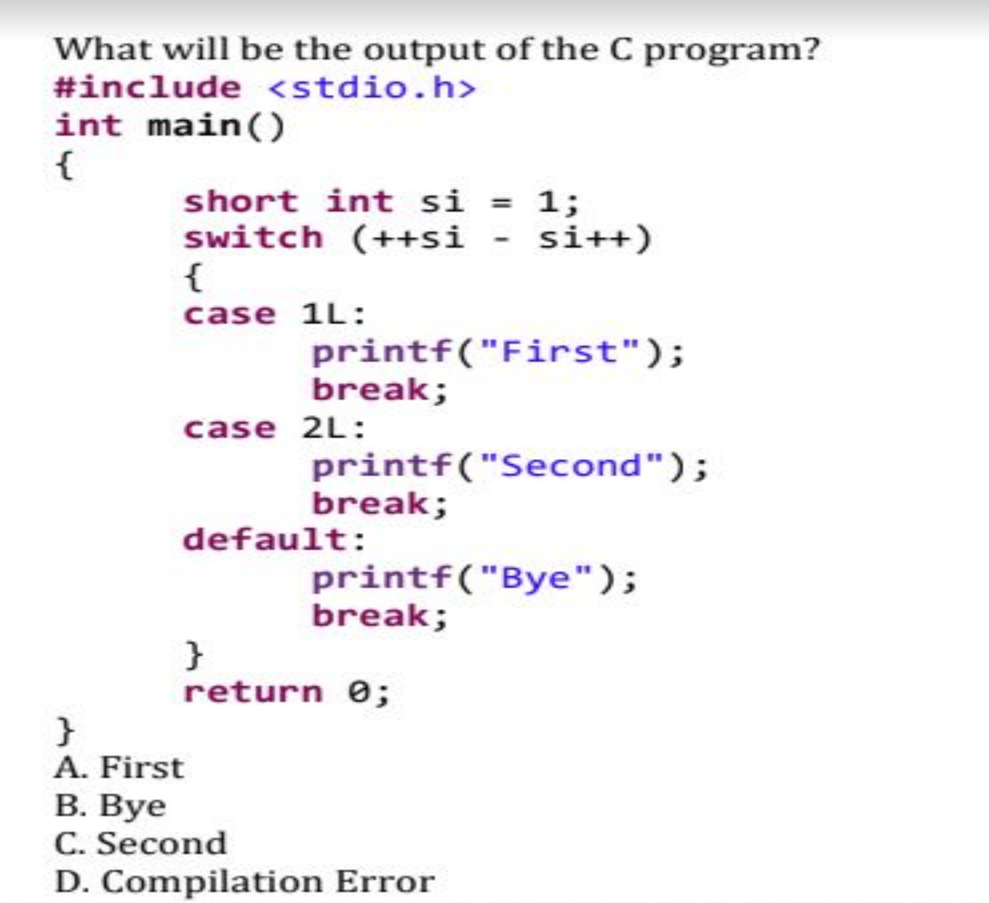
What will be the output of the C program?

```
#include <stdio.h>
#define N 1.5
int main()
{
        int i = 6;
        switch (i)
        {
        case N + 5:
                printf("Switch ON the fan");
                break;
        case N * 4:
                printf("Switch ON the Air Cooler");
                break;
        default:
                printf("Save Energy");
        }
        return 0;
}
```

A. Switch ON the fan
B. Switch ON the Air Cooler
C. Save Energy
D. Compilation Error

ANSWER:D

MCQS



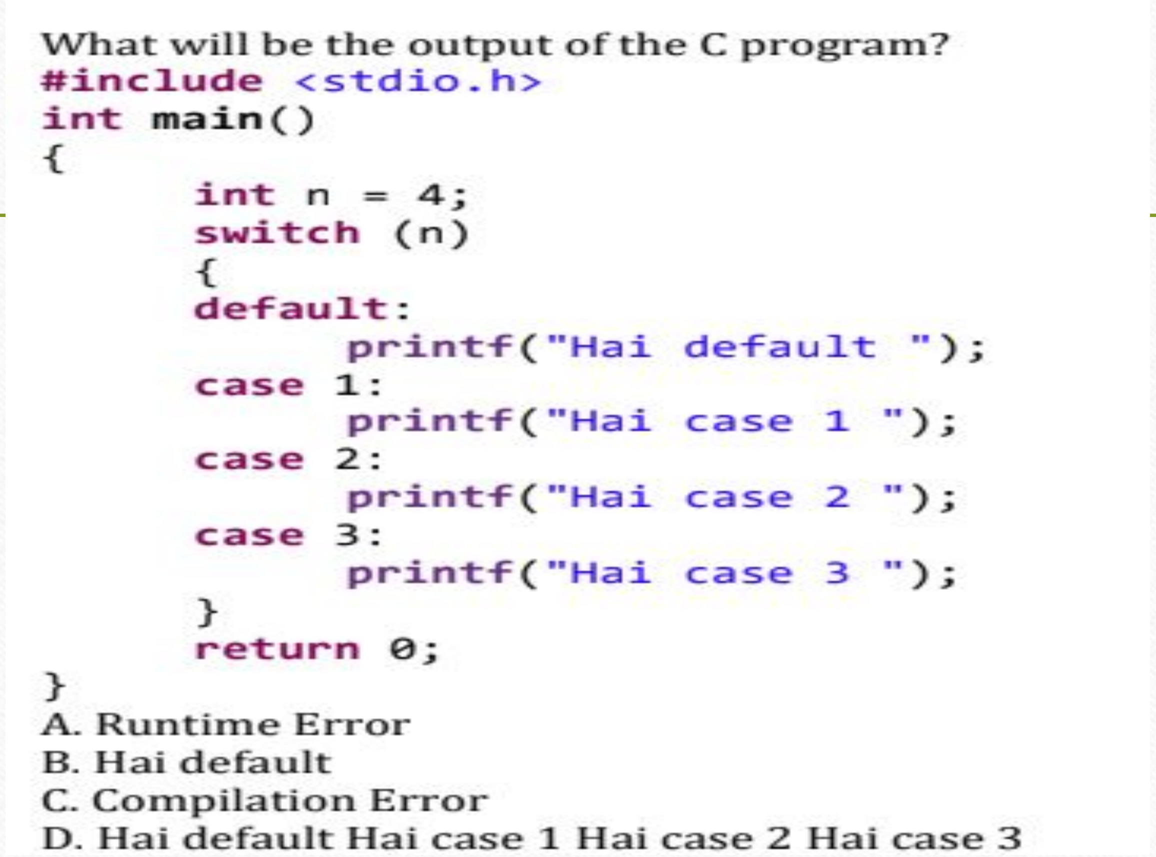
What will be the output of the C program?

```
#include <stdio.h>
int main()
{
        short int si = 1;
        switch (++si - si++)
        {
        case 1L:
                printf("First");
                break;
        case 2L:
                printf("Second");
                break;
        default:
                printf("Bye");
                break;
        }
        return 0;
}
```

A. First
B. Bye
C. Second
D. Compilation Error

ANSWER:B

MCQS



What will be the output of the C program?

```
#include <stdio.h>
int main()
{
        int n = 4;
        switch (n)
        {
        default:
                printf("Hai default ");
        case 1:
                printf("Hai case 1 ");
        case 2:
                printf("Hai case 2 ");
        case 3:
                printf("Hai case 3 ");
        }
        return 0;
}
```

A. Runtime Error
B. Hai default
C. Compilation Error
D. Hai default Hai case 1 Hai case 2 Hai case 3

ANSWER:D

Thank You