1. Given a maximum of 100 digit numbers as input, find the difference between the sum of odd and even position digits

```
import java.until.Scanner;

public class Ex=2{

public static void main(string[]args){

Scanner s=new Scanner(System.in);

long a=s.nextLong();

long temp=a;

int flag=0,even=0,odd=0;

string aa=string.valueof(temp);

int len=aa.length();

while(a>0)

{

long rem=a%10;

a/=10;

if(len%2==0)

{

if(flag==0)

{

even+=rem;

flag=1;

}

else if(flag==1)

{
```

```
odd+=rem;
flag=0;
}
}
else
{
if(flag==0)
{
odd+=rem;
flag=1;
}
else if(flag==1)
{
even+=rem;
flag=0;
}
}
}
int sum=odd-even;
if(sum<0)
{
sum=-sum;
}
```

System.out.println(sum);

}

}

Test Cases

Case 1

Input: 4567

Expected Output: 2

Explanation : Odd positions are 4 and 6 as they are pos: 1 and pos: 3, both have sum 10. Similarly, 5 and 7 are at even positions pos: 2 and pos: 4 with sum 12. Thus, difference is 12 – 10 = 2

Case 2

Input: 5476

Expected Output: 2

Case 3

Input: 9834698765123

Expected Output: 1

2. One programming language has the following keywords that cannot be used as identifiers:

   break, case, continue, default, defer, else, for, func, goto, if, map, range, return, struct, type, var

   Write a program to find if the given word is a keyword or not

```
#include<stdio.h>

#include<string.h>

Int main(){

Char
str[16][10]={“break”,”case”,”continue”,”default”,”defer”,”else”,”for”,”func”,”goto”,”if”,”map”,
”range”,”return”,”struct”,”type”,”var”};
```

```
Char input[20];
int flag=0;
        scanf("%s",input);
        for(int i=0;i<16;i++){
        if(strcmp(input,str[i])==0){
        flag =1;
        break;
        }
        }
        If(flag==1){
        Printf("%s" is a keyword",input);
        }
        else{
        printf("%s is not a keyword",input);
        }
        return 0;
        }
```

Test cases

Case 1

Input – defer

Expected Output – defer is a keyword

Case 2

Input – While

Expected Output – while is not a keyword

3. **Consider the below series:**

1, 2, 1, 3, 2, 5, 3, 7, 5, 11, 8, 13, 13, 17.....

This series is a mixture of 2 series fail the odd terms in this series form a Fibonacci series and all the even terms are the prime numbers in ascending order

Write a program to find the Nth term in this series

The value N in a positive integer that should be read from mm. The Nth term that is calculated by the program should be written to STDOUT Otherthan the value of Nth term , no other characters / string or message should be written to STDOUT.

For example, when N:14, the 14th term in the series is 17 So only the value 17 should be printed to STDOUT

```
#include<stdio.h>
#define MAX 1000
void fibonacci(int n)
{
 int i, term1 = 0, term2 = 1, nextTerm;
 for (i = 1; i<=n; i++)
 {
  nextTerm = term1 + term2;
  term1 = term2;
  term2 = nextTerm;
 }
 printf("%d", term1);
}

void prime(int n)
{
 int i, j, flag, count =0;
 for (i=2; i<=MAX; i++)
 {
  flag = 0;
  for (j=2; j<i; j++)
  {
   if(i%j == 0)
   {
    flag = 1;
    break;
   }
  }
  if (flag == 0)
   count++;
  if(count == n)
```

```
  {
   printf("%d", i);
   break;
  }
 }
}
int main( )
{
 int n;
 scanf("%d", &n);
 if(n%2 == 1)
  fibonacci (n/2 + 1);
 else
  prime(n/2);
 return 0;
}
```