Given a positive integer denoting , do the following: If , print the lowercase English word corresponding to the number (e.g., one for , two for , etc.). If , print Greater than 9.

#include <assert.h>

#include <limits.h>

#include <math.h>

#include <stdbool.h>

#include <stddef.h>

#include <stdint.h>

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

char\* readline();

int main()

{

    char\* n\_endptr;

    char\* n\_str = readline();

    int n = strtol(n\_str, &n\_endptr, 10);

    scanf("%d", &n);

    if (n >= 1 && n <= 9) {

        if (n == 1)

            printf("one\n");

        else if (n == 2)

            printf("two\n");

        else if (n == 3)

            printf("three\n");

        else if (n == 4)

            printf("four\n");

        else if (n == 5)

            printf("five\n");

        else if (n == 6)

            printf("six\n");

        else if (n == 7)

            printf("seven\n");

        else if (n == 8)

            printf("eight\n");

        else if (n == 9)

            printf("nine\n");

    } else {

        printf("Greater than 9\n");

    }

    if (n\_endptr == n\_str || \*n\_endptr != '\0') { exit(EXIT\_FAILURE); }

    // Write Your Code Here

    return 0;

}

char\* readline() {

    size\_t alloc\_length = 1024;

    size\_t data\_length = 0;

    char\* data = malloc(alloc\_length);

    while (true) {

        char\* cursor = data + data\_length;

        char\* line = fgets(cursor, alloc\_length - data\_length, stdin);

        if (!line) { break; }

        data\_length += strlen(cursor);

        if (data\_length < alloc\_length - 1 || data[data\_length - 1] == '\n') { break; }

        size\_t new\_length = alloc\_length << 1;

        data = realloc(data, new\_length);

        if (!data) { break; }

        alloc\_length = new\_length;

    }

    if (data[data\_length - 1] == '\n') {

        data[data\_length - 1] = '\0';

    }

    data = realloc(data, data\_length);

    return data;

}

**Task**

For each integer  in the interval  (given as input) :

* If , then print the English representation of it in lowercase. That is "one" for , "two" for , and so on.
* Else if  and it is an even number, then print "even".
* Else if  and it is an odd number, then print "odd"

#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>

int main()

{

        int a, b;

    scanf("%d\n%d", &a, &b);

      // Complete the code.

    for(int i=a;i<=b;i++){

        if(i>=a && i<=b)

        {

                    if(i == 1){

                printf("one\n");

            }

            else if(i == 2){

                printf("two\n");

            }

            else if(i == 3){

                printf("three\n");

            }

            else if (i == 4) {

                printf("four\n");

            }

            else if (i == 5) {

                printf("five\n");

            }

            else if(i == 6){

                printf("six\n");

            }

            else if(i == 7){

                printf("seven\n");

            }

            else if(i == 8){

                printf("eight\n");

            }

            else if (i == 9) {

                printf("nine\n");

            }

            else{

                if(i % 2 == 0){

                    printf("even\n");

                }

                else{

                    printf("odd\n");

                }

            }

        }

    }

    return 0;

}

Given a five digit integer, print the sum of its digits.

#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>

int main() {

    int n,no,d,sum=0;

    scanf("%d", &n);

    //Complete the code to calculate the sum of the five digits on n.

    if(n>= 10000 && n<=99999){

        no=n;

        while(no != 0){

            d = no %10;

            sum = sum + d;

            no = no /10;

        }

        printf("%d",sum);

    }

    else{

        printf("kindly enter a five digit number");

    }

    return 0;

}

Complete the *calculate\_the\_maximum* function in the editor below.

*calculate\_the\_maximum* has the following parameters:

* *int n:* the highest number to consider
* *int k:* the result of a comparison must be lower than this number to be considered

#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>

//Complete the following function.

void calculate\_the\_maximum(int n, int k) {

  //Write your code here.

  int maxand=0,maxor=0,maxeor=0,a,b;

  for (a=1;a<=n;a++)

    {

        for(b=a+1;b<=n;b++)

        {

            int uand=a & b;

            int uor= a | b;

            int uxor = a ^ b;

            if(uand < k && uand >maxand)

            {

                maxand=uand;

            }

             if(uor < k && uor >maxor)

            {

                maxor=uor;

            }

             if(uxor < k && uxor >maxeor)

            {

                maxeor=uxor;

            }

        }

    }

    printf("%d\n%d\n%d\n",maxand,maxor,maxeor);

}

int main() {

    int n, k,a,b;

    scanf("%d %d", &n, &k);

    calculate\_the\_maximum(n, k);

    return 0;

}

Array reverse

#include <stdio.h>

#include <stdlib.h>

int main()

{

    int num, \*arr, i;

    //printf("enter no of elements : ");

    scanf("%d", &num);

    arr = (int\*) malloc(num \* sizeof(int));

    for(i = 0; i < num; i++) {

        scanf("%d", arr + i);

    }

    /\* Write the logic to reverse the array.

    for(i = 0; i < num; i++)

        printf("%d ", \*(arr + i));

     \*/

    int \*arr2,j=0;

    arr2=(int\*) malloc(num \* sizeof(int));

    for(int i= num-1;i>=0;i--)

    {

        arr2[j]=arr[i];

        j++;

    }

     for(i = 0; i < num; i++)

        printf("%d ", \*(arr2 + i));

    return 0;

}

String new line

#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>

int main() {

    char \*s;

    s = malloc(1024 \* sizeof(char));

    scanf("%[^\n]", s);

    s = realloc(s, strlen(s) + 1);

    //Write your logic to print the tokens of the sentence here.

    int len=strlen(s);

    for(int i=0;i<len;i++)

    {

        if(s[i]==' ')

        {

            printf("\n");

        }

        else {

            printf("%c",s[i]);

        }

    }

    return 0;

}

Digit count in a string

#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>

int main() {

    /\* Enter your code here. Read input from STDIN. Print output to STDOUT \*/

    char \*str;

    str = malloc(1024 \* sizeof(char));

    scanf("%[^\n]", str);

    str = realloc(str, strlen(str) + 1);

    int count[10]={0};

    int len=strlen(str);

    for(int i=0;i<len;i++)

    {

        if(str[i]>='0' && str[i]<='9')

        {

            int c=str[i]-'0';

            count[c]++;

        }

    }

    for(int i=0;i<10;i++)

    {

        printf("%d ",count[i]);

    }

    return 0;

}

Marks by gender

#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>

//Complete the following function.

int marks\_summation(int\* marks, int number\_of\_students, char gender) {

  //Write your code here.

  int sum=0;

  if(gender=='b')

  {

      for (int i = 0; i < number\_of\_students; i += 2)

        {

            sum += marks[i];

        }

  }

  else if (gender =='g') {

  for (int i = 1; i < number\_of\_students; i += 2)

        {

            sum += marks[i];

        }

  }

  return sum;

}

int main() {

    int number\_of\_students;

    char gender;

    int sum;

    scanf("%d", &number\_of\_students);

    int \*marks = (int \*) malloc(number\_of\_students \* sizeof (int));

    for (int student = 0; student < number\_of\_students; student++) {

        scanf("%d", (marks + student));

    }

    scanf(" %c", &gender);

    sum = marks\_summation(marks, number\_of\_students, gender);

    printf("%d", sum);

    free(marks);

    return 0;

}

Nth number

#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>

//Complete the following function.

int find\_nth\_term(int n, int a, int b, int c) {

  //Write your code here.

 if( n == 1)

  return a;

 if (n == 2)

 return b;

 if(n == 3)

  return c;

  return find\_nth\_term(n-1,  a, b, c)+find\_nth\_term(n-2,  a, b, c)+find\_nth\_term(n-3,  a, b, c);

}

int main() {

    int n, a, b, c;

    scanf("%d %d %d %d", &n, &a, &b, &c);

    int ans = find\_nth\_term(n, a, b, c);

    printf("%d", ans);

    return 0;

}