Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра информатики

Дисциплина: Избранные главы информатики

# ОТЧЕТ
к лабораторной работе
на тему

# Стандартные типы данных, коллекции, функции, модули в Python

БГУИР КП 1-40 04 01

Выполнила: студентка группы 253502,
Меликова Камилла Раидовна

Проверила: Жвакина Анна Васильевна

**Задание 1.**

В соответствии с заданием своего варианта составить программу для вычисления значения функции с помощью разложения функции в степенной ряд. Задать точность вычислений eps.

Предусмотреть максимальное количество итераций, равное 500.

Вывести количество членов ряда, необходимых для достижения указанной точности вычислений. Результат получить в виде таблицы.

14. $$\ln(1-x)=\sum_{n=0}^{\infty}(-1)\frac{x^n}{n}=-x-\frac{x^2}{2}-\frac{x^3}{3}+\ldots,\ |x|<1$$

**Решение:**

Модуль Input:

```
def input_x(x):
    """Returns x value."""
    while True:
        try:
            x = float(input("Enter the value of x:"))
            if abs(x)>=1:
                raise ValueError("x doesn't suit conditions!")
            break;
        except ValueError as err:
            print(err)
            print("Incorrect input, try again!")
    return x


def input_accuracy(eps):
    """Returns epsilon value."""
    while True:
        try:
            eps=float(input("Enter the accuracy:"))
            if(eps<=0):
                raise ValueError("accuracy must be positive!")
            if(eps>=1):
                raise ValueError("accuracy must be less than 1!")
            break;
        except ValueError as err:
            print(err)
            print("Incorrect input, try again!")
    return eps
```

```python
def continue_or_exit():
    """Provides a choice to the user to continue or exit the program."""
    while True:
        choice = input("Would you like to continue (c) or exit (e)? ").strip().lower()
        if choice == 'c':
            return True
        elif choice == 'e':
            return False
        else:
            print("Invalid choice. Please enter 'C' to continue or 'E' to exit.")
```

Модуль Output:

```python
import math

def invitation():
    """Outputs the invitation to start the program at the beginning."""
    print("This program is designed to calculate the amount of iterations"
          " required for a numerical series to converge.")


def print_table(x, n, sum, y, eps):
    """Outputs the result of the program in a table."""
    amount_of_digits = int(abs(math.log10(eps)))
    print("{:^15} | {:^15} | {:^20} | {:^20} | {:^20}".format("x", "n", "F(x)", "Math F(x)", "eps"))
    print("-" * 99)
    print("{:^15} | {:^15d} | {:^20.{}f} | {:^20.{}f} | {:.{}f}".format(x, n, sum, amount_of_digits, y, amount_of_digits,
```

Модуль Decorators:

```python
def decorate(func):
    """Decorates given function."""
    def new_func(x):
        print("Running function:",func.__name__)
        print("Arguments:",x)
        res=func(x)
        print("Result:",res)
        return res
    return new_func
```

Модуль Calculations:

```python
import math
from Decorators import decorate

@decorate
def get_function_value(x):
    """Calculates the function ln(1-x).

    Arguments:
    x -- float argument,|x|<1
    """
    return math.log(1-x)
```

```python
def get_series_sum(x,eps,y):
    """Calculates the sum of given series.

    Arguments:
    x -- float argument
    eps -- specified accuracy
    y -- function's value

    Returns the sum and amount of iterations,
    alerts, if accuracy hasn't been achieved.
    """
    sum,component,n,prev_result=0,0,1,1000
    while True:
        component=-1*x**n/n
        sum+=component
        n+=1
        if n==501:
            print("500 iterations already done.")
            return sum,n;
        if(abs(sum-y)<eps):
            print("Accuracy is achieved.")
            return sum,n
        elif prev_result<=abs(sum-y):
            print("Defined accuracy can't be achieved.")
            return sum,n
        prev_result=abs(sum-y)
```

Main:

```python
# Program Purpose: Task 1.
# Lab Work Number: 3
# Program Name: Convergence of Numerical Series
# Version: 1.0
# Developer: Melikava Kamila
# Date: 25.03.2024

import Input as inpt
from Output import invitation, print_table
import Calculation as calc


invitation()    #Print invitation at the beginning of the program.

while True:
    x=inpt.input_x(None)        #Input argument x.
    print("You entered x =",x)

    eps=inpt.input_accuracy(None)   #Input argument epsilon.
    print("You entered epsilon =",eps)

    y=calc.get_function_value(x)    #Calculations.
    sum,n=calc.get_series_sum(x,eps,y)

    print_table(x,n,sum,y,eps)  #Output the result table.

    if not inpt.continue_or_exit(): #Ask user if he wants to finish the program or continue.
        break
```

Результат работы программы:

```
This program is designed to calculate the amount of iterations required for a numerical series to converge.
Enter the value of x:hfjh
could not convert string to float: 'hfjh'
Incorrect input, try again!
Enter the value of x:23
x doesn't suit conditions!
Incorrect input, try again!
Enter the value of x:0.0001
You entered x = 0.0001
Enter the accuracy:erfojf
could not convert string to float: 'erfojf'
Incorrect input, try again!
Enter the accuracy:34
accuracy must be less than 1!
Incorrect input, try again!
Enter the accuracy:1e-15
You entered epsilon = 1e-15
Running function: get_function_value
Arguments: 0.0001
Result: -0.00010000500033334732
Accuracy is achieved.
       x        |        n        |         F(x)         |      Math F(x)       |         eps
------------------------------------------------------------------------------------------------
    0.0001      |        4        |  -0.000100005000333  |  -0.000100005000333  | 0.000000000000001
Would you like to continue (c) or exit (e)?
```

Программа проверяет вводимые данные на корректность, выводит решение, а также предоставляет возможность выбора продолжения или прерывания программы. В программе также используется декоратор.

**Задание 2**.

Организовать цикл, который принимает целые числа и вычитает их из 10000. Окончание – получение отрицательного итога. Цель – найти сумму последовательности чисел.

Модуль Input:

```python
def input_int():
    """Returns integer value."""
    while True:
        try:
            x = int(input("Enter the number:"))
            break;
        except ValueError:
            print("Incorrect input, try again!")
    return x
```

```
def generate_random_sequence():
    """Generator for a sequence of floating-point values."""
    while True:
        yield random.randint(0, 15000)


def continue_or_exit():
    """Provides a choice to the user to continue or exit the program."""
    while True:
        choice = input("Would you like to continue (c) or exit (e)? ").strip().lower()
        if choice == 'c':
            return True
        elif choice == 'e':
            return False
        else:
            print("Invalid choice. Please enter 'C' to continue or 'E' to exit.")
```

Модуль Operation:

```
def perform_operation(a):
    """Subtracts a from 100000."""
    return 10000-a
```

Main:

```
import Input as inp
import Operation as op

print("This program expects an integer value input and subtracts it from 10000."
      "The cycle breaks if result is negative.")
numbers_sum=0
while True:
        print("Choose how do you want to input numbers' sequence:\n"
          "1. Manually.\n"
          "2. Generated automatically.\n")

        while True:
            print("Enter 1 or 2:")
            choice=inp.input_int()
            if choice!=1 and choice!=2:
                print("Incorrect input!Try again.")
            else:
                break

        if choice==1:
            while True:
                a=inp.input_int()   #Input an integer value
                numbers_sum+=a
                res=op.perform_operation(a) #Substract a from 1000
                print("Result:",res)
                if res<0:   #If result is negative break the cycle
                    break
```

```
        elif choice==2:
            for num in inp.generate_random_sequence():
                print(num)
                numbers_sum+=num
                res=op.perform_operation(num)
                print("Result:",res)
                if res<0:   #If result is negative break the cycle
                    break

        print("Sum of numbers is",numbers_sum);
        numbers_sum=0

        if not inp.continue_or_exit(): #Ask user if he wants to finish the program or continue.
            break
```

Результат работы программы:

```
This program expects an integer value input and subtracts it from 10000.The cycle breaks if result is negative.
Choose how do you want to input numbers' sequence:
1. Manually.
2. Generated automatically.

Enter 1 or 2:
Enter the number:апкп
Incorrect input, try again!
Enter the number:4
Incorrect input!Try again.
Enter 1 or 2:
Enter the number:1
Enter the number:2
Result: 9998
Enter the number:3
Result: 9997
Enter the number:4
Result: 9996
Enter the number:100001
Result: -90001
Sum of numbers is 100010
Would you like to continue (c) or exit (e)? c
Invalid choice. Please enter 'C' to continue or 'E' to exit.
Would you like to continue (c) or exit (e)? c
Choose how do you want to input numbers' sequence:
1. Manually.
2. Generated automatically.

Enter 1 or 2:
Enter the number:2
5772
Result: 4228
4455
Result: 5545
13466
Result: -3466
Sum of numbers is 23693
Would you like to continue (c) or exit (e)? e
Для продолжения нажмите любую клавишу . . .
```

Программа обрабатывает некорректный пользовательский ввод, позволяет продолжить программу или окончательно завершить. Также предоставлен выбор для ввода числовой последовательности: пользователь может ввести ее вручную либо получить сгенерированную автоматически с использованием функции генератора.

**Задание 3.**
В строке, вводимой с клавиатуры, подсчитать количество пробельных символов и запятых.

Решение:

Модуль Input:

```
def continue_or_exit():
    """Provides a choice to the user to continue or exit the program."""
    while True:
        choice = input("Would you like to continue (c) or exit (e)? ").strip().lower()
        if choice == 'c':
            return True
        elif choice == 'e':
            return False
        else:
            print("Invalid choice. Please enter 'C' to continue or 'E' to exit.")
```

Модуль CountSymbols:

```
def count_symbols(s,s1,s2):
    """Counts commas' and spaces' amount in a given string."""
    return s.count(s1),s.count(s2)
```

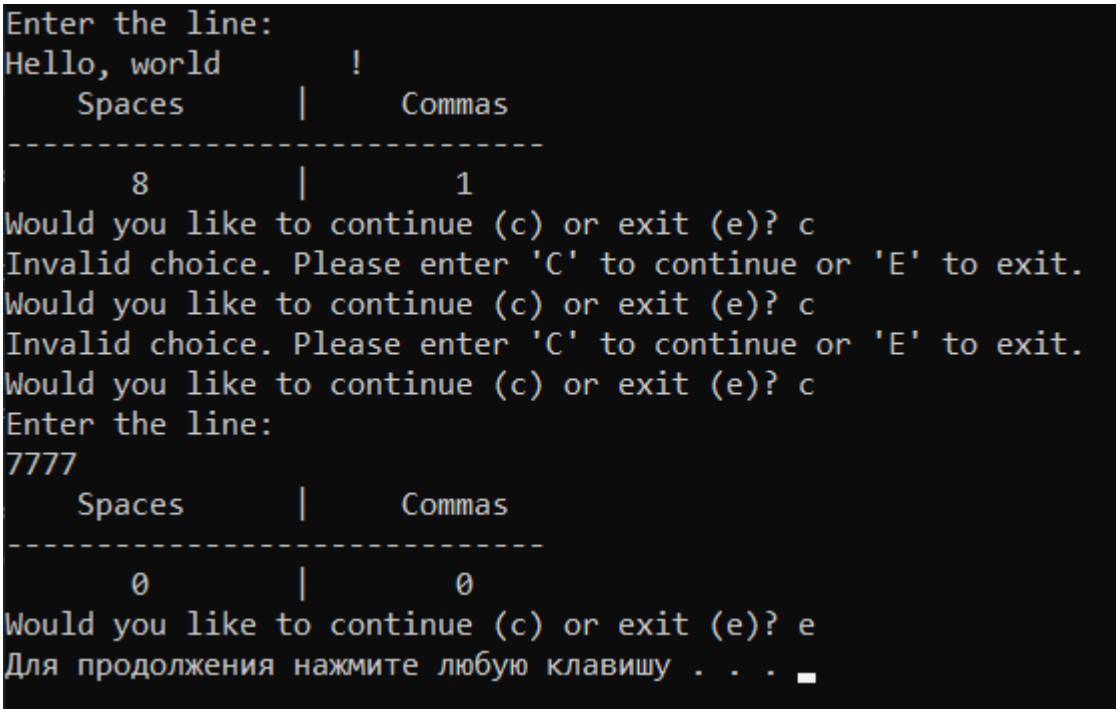
Модуль Output:

```
def print_result(res1, res2):
    """Outputs the result of the program in a table."""
    print("{:^15} | {:^15}".format("Spaces", "Commas"))
    print("-" * 30)
    print("{:^15d} | {:^15d}".format(res1,res2))
```

Main:

```
# Program Purpose: Task 3.
# Lab Work Number: 3
# Program Name: Counter of spaces and commas
# Version: 1.0
# Developer: Melikava Kamila
# Date: 25.03.2024

from CountSymbols import count_symbols
from Output import print_result
from Input import continue_or_exit

while True:
    print("Enter the line:")
    s=str(input())
    s1=" "  #Specify first symbol.
    s2=","  #Specify second symbol.

    res1,res2=count_symbols(s,s1,s2)    #Count symbols.

    print_result(res1,res2)    #Output the result in a table.

    if not continue_or_exit():  #Ask user if he wants to finish the program or continue.
        break
```

Результат  работы программы:




Данная программа выводит результат в виде таблицы, предлагает продолжение или завершение работы.

**Задание 4.**

Дана строка текста, в которой слова разделены пробелами и запятыми. В соответствии с заданием своего варианта составьте программу для анализа строки, инициализированной в коде программы:

«So she was considering in her own mind, as well as she could, for the hot day made her feel very sleepy and stupid, whether the pleasure of making a daisy-chain would be worth the trouble of getting up and picking the daisies, when suddenly a White Rabbit with pink eyes ran close by her.»

Если не оговорено иное, то регистр букв при решении задачи не имеет значения.

Вариант:

а) определить число слов, длина которых меньше 7 символов;
б) найти самое короткое слово, заканчивающееся на букву 'a';
в) вывести все слова в порядке убывания их длин

Модуль Input:

```
def input_int():
    """Returns integer value."""
    while True:
        try:
            x = int(input("Enter the number from 1 to 3:"))
            if x!=1 and x!=2 and x!=3:
                raise ValueError("choice doesn't suit conditions!")
            break;
        except ValueError:
            print("Incorrect input, try again!")
    return x

def continue_or_exit():
    """Provides a choice to the user to continue or exit the program."""
    while True:
        choice = input("Would you like to continue (c) or exit (e)? ").strip().lower()
        if choice == 'c':
            return True
        elif choice == 'e':
            return False
        else:
            print("Invalid choice. Please enter 'C' to continue or 'E' to exit.")
```

Модуль LexicalAnalysis:

```
def get_words(line):
    """Splits a line into words by spaces and delimeters."""
    separators = [',', '.']
    for separator in separators:
        line = ' '.join(line.split(separator))
    return line.split()

def get_short_words(words):
    """Returns words whose length is less than 7."""
    return [word for word in words if len(word)<7]

def get_shortest_a_word(words):
    """Returns the shortest word ending with 'a'."""
    finishing_on_a=[word for word in words if word.endswith('a')]
    return min(finishing_on_a, key=len)

def get_sorted_words(words):
    """Returns list of words sorted by words' length in reversed order."""
    return sorted(words,key=lambda word:len(word),reverse=True)
```

Модуль Output:

```
def print_list(words):
    """Outputs the given list"""
    for word in words: print(word)

def message():
    """"Describes program's functionality."""
    print("This program analyzes given string and outputs following data about it:\n"
          "1. Get the list of words whose length is less than 7.\n"
          "2. Get the shortest word ending with 'a'.\n"
          "3. Get sorted list of words.")
```

Main:

```
import LexicalAnalysis as analyzer
import Output as out
import Input as inp

given_str="So she was considering in her own mind, as well as she could," \
          " for the hot day made her feel very sleepy and stupid, whether the pleasure" \
          " of making a daisy-chain would be worth the trouble of getting up and picking the daisies," \
          " when suddenly a White Rabbit with pink eyes ran close by her."


words=analyzer.get_words(given_str)  #Split line into words.
out.message()  #Print the program's functionality.
while True:
    choice = inp.input_int()  #Input function's number.

    if choice==1:
        print("The list of words whose length is less than 7:")
        short_words=analyzer.get_short_words(words)
        out.print_list(short_words)
    elif choice==2:
        print("The shortest word ending with a is:")
        word=analyzer.get_shortest_a_word(words)
        print(word)
    elif choice==3:
        print("Words sorted in reversed order by their length:")
        sorted_words=analyzer.get_sorted_words(words)
        out.print_list(sorted_words)
    if not inp.continue_or_exit():  #Ask user if he wants to finish the program or continue.
        break
```

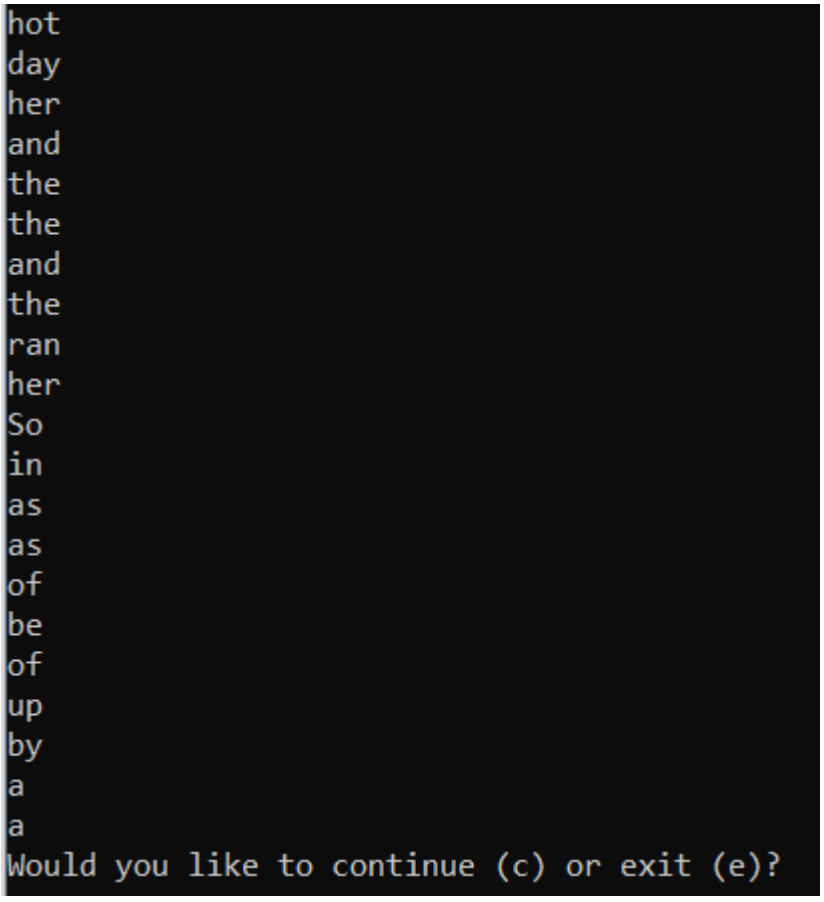
Результат работы программы:

```
This program analyzes given string and outputs following data about it:
1. Get the list of words whose length is less than 7.
2. Get the shortest word ending with 'a'.
3. Get sorted list of words.
Enter the number from 1 to 3:1
The list of words whose length is less than 7:
So
she
was
in
her
own
mind
as
well
as
she
could
for
the
hot
day
made
```

```
her
feel
very
sleepy
and
stupid
the
of
making
a
would
be
worth
the
of
up
and
the
```

```
Would you like to continue (c) or exit (e)? c
Enter the number from 1 to 3:2
The shortest word ending with a is:
a
```

```
Enter the number from 1 to 3:3
Words sorted in reversed order by their length:
considering
daisy-chain
pleasure
suddenly
whether
trouble
getting
picking
daisies
sleepy
stupid
making
Rabbit
could
would
worth
White
close
mind
well
made
```


Данная программа выполняет задание по варианту для заданной строки, предлагает продолжить или завершить работу.

**Задание 5.**

В соответствии с заданием своего варианта составить программу для обработки вещественных списков. Программа должна содержать следующие базовые функции:

1) ввод элементов списка пользователем;

2) проверка корректности вводимых данных;

3) реализация основного задания с выводом результатов;

4) вывод списка на экран.

Вариант 14:

| |
|---|
| Найти номер минимального отрицательного элемента списка и сумму элементов списка, расположенных между первым и вторым отрицательными элементами |

Модуль Input:

```
def input_positive(line):
    """Returns positive integer value."""
    while True:
        try:
            x = int(input(line))
            if(x<=0):
                raise ValueError("It can't be negative!")
            break;
        except ValueError as err:
            print(err)
            print("Incorrect input, try again!")
    return x


def input_int():
    """Returns integer value."""
    while True:
        try:
            x = int(input("Enter the number:"))
            break;
        except ValueError:
            print("Incorrect input, try again!")
    return x


def input_float():
    """Returns float value."""
    while True:
        try:
            x = float(input("Enter the float number:"))
            break;
        except ValueError:
            print("Incorrect input, try again!")
    return x
```

```python
def generate_random_sequence(length):
    """Generator for a sequence of floating-point values."""
    for _ in range(length):
        yield random.uniform(-100, 100)


def manual_input(length):
    """Provides a manual input of list's elements."""
    lst=[]
    while length!=0:
        elem=input_float()
        lst.append(elem)
        length-=1
    return lst


def continue_or_exit():
    """Provides a choice to the user to continue or exit the program."""
    while True:
        choice = input("Would you like to continue (c) or exit (e)? ").strip().lower()
        if choice == 'c':
            return True
        elif choice == 'e':
            return False
        else:
            print("Invalid choice. Please enter 'C' to continue or 'E' to exit.")
```

Модуль Output:

```python
def message():
    pass

def print_list(lst):
    """Outputs the given list"""
    for word in lst: print(word)

def output_value(value,line):
    """Outputs task's results."""
    if value==None:
        print("There is nothing to output.")
    else:
        print(line,value)
```

Модуль TaskVariant:

```python
def find_min_negative_index(lst):
    """Finds the index of the least negative element."""
    neg_index = None  #neg_index is set to None because we can have a list without negative numbers.
    for i, elem in enumerate(lst):
        if elem < 0:
            if neg_index is None or lst[i] < lst[neg_index]:  #Update neg_index value.
                neg_index = i
    return neg_index
```

```python
def sum_between_negatives(lst):
    """Finds the sum of elements between first two negative elements."""
    neg_index_1 = None
    neg_index_2 = None
    for i, elem in enumerate(lst):
        if elem < 0:
            if neg_index_1 is None:
                neg_index_1 = i
            elif neg_index_2 is None:
                neg_index_2 = i
                break
    if neg_index_1 is None or neg_index_2 is None or (neg_index_2-neg_index_1)==1:
        return None
    else:
        return sum(lst[neg_index_1+1:neg_index_2])
```

Main:

```python
import Input as inp
import Output as out
import TaskVariant as task

while True:
    length=inp.input_positive("Enter list's length:")

    print("Choose how do you want to input list's elements:\n"
          "1. Manually.\n"
          "2. Generated automatically.\n")

    while True:
        choice=inp.input_positive("Enter 1 or 2:")
        if choice!=1 and choice!=2:
            print("Incorrect input!Try again.")
        else:
            break

    lst=[]
    if choice==1:
        lst=inp.manual_input(length)
        print("You successfully entered the list:")
    elif choice==2:
        random_sequence = inp.generate_random_sequence(length)
        lst = list(random_sequence)
        print("List has been generated:")

    out.print_list(lst)

    index=task.find_min_negative_index(lst)
    out.output_value(index,"The least negative number's index is")

    sum=task.sum_between_negatives(lst)
    out.output_value(sum,"The sum of elements between first two negatives:")

    if not inp.continue_or_exit():
        break;
```

Результат работы программы:

```
Enter list's length:пка
invalid literal for int() with base 10: 'пка'
Incorrect input, try again!
Enter list's length:-1
It can't be negative!
Incorrect input, try again!
Enter list's length:10
Choose how do you want to input list's elements:
1. Manually.
2. Generated automatically.

Enter 1 or 2:2
List has been generated:
-10.467227661938878
95.83465140795414
74.59349792143172
-17.711741272278488
-14.904757639804387
-16.505020698353917
-73.18556089087791
-61.145764629565576
-98.60574163254476
-44.95477330443223
The least negative number's index is 8
The sum of elements between first two negatives: 170.42814932938586
Would you like to continue (c) or exit (e)? c
```

```
Enter list's length:3
Choose how do you want to input list's elements:
1. Manually.
2. Generated automatically.

Enter 1 or 2:1
Enter the float number:2
Enter the float number:2
Enter the float number:2
You successfully entered the list:
2.0
2.0
2.0
There is nothing to output.
There is nothing to output.
Would you like to continue (c) or exit (e)? c
Enter list's length:4
Choose how do you want to input list's elements:
1. Manually.
2. Generated automatically.

Enter 1 or 2:-1
It can't be negative!
Incorrect input, try again!
Enter 1 or 2:1
Enter the float number:-4
Enter the float number:-4
Enter the float number:-5
Enter the float number:-6
You successfully entered the list:
-4.0
-4.0
-5.0
-6.0
The least negative number's index is 3
There is nothing to output.
Would you like to continue (c) or exit (e)? e
Для продолжения нажмите любую клавишу . . .
```

Программа обрабатывает некорректный пользовательский ввод, предлагает на выбор два варианта ввода списка: вручную и с помощью генератора. Размер списка задается пользователем. В случае если между отрицательными числами нет чисел, выводится соответствующее

сообщение. Также предлагается на выбор продолжить или завершить программу.