

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 №3  
на тему:  
**«СТАНДАРТНЫЕ ТИПЫ ДАННЫХ, КОЛЛЕКЦИИ, ФУНКЦИИ,  
МОДУЛИ»**  
БГУИР 6-05-0612-02 24

Выполнил студент группы 353501  
СОРОКА Евгений Викторович

---

(дата, подпись студента)

Проверил доцент кафедры информатики  
ЖВАКИНА Анна Васильевна

---

(дата, подпись преподавателя)

Минск 2025

Код программы представлен ниже.

### Menu.py

```
import input_checker import task1 import task2 import task3 import task4
import task5
def main(): """ Main function to choose task num. """
    while True:
        print("==MENU==")
        print("Task 1")
        print("Task 2")
        print("Task 3")
        print("Task 4")
        print("Task 5")
        print("Exit (0)")

        user_choice = input_checker.input_checker("Input num of task or 0 to exit
", int, 0, 5)

        if user_choice == 1:
            task1.task1()

        elif user_choice == 2:
            task2.task2()

        elif user_choice == 3:
            task3.task3()

        elif user_choice == 4:
            task4.task4()

        elif user_choice == 5:
            task5.task5()

        elif user_choice == 0:
            print("Exit")
            break

    if name == "main": main()
```

### Input\_checker.py

```
import random, string
def input_checker(prompt, type_of_data, min_value = None, max_value =
None): """ This function check user input for tasks.
:param prompt: user's input
:param type_of_data: type of user prompt data
:param min_value: minimum required value for task
:param max_value: maximum required value for task
:return: valid user's input OR error
"""

while True:
```

```

try:
    users_input = type_of_data(input(prompt))

    if min_value is not None and users_input < min_value:
        raise ValueError(f"Input value greater or equal than
{min_value}")

    elif max_value is not None and users_input > max_value:

        raise ValueError(f"Input value lower or equal than {min_value}")
    return users_input;

except ValueError:
    print(f"Errors : {ValueError}. Please enter one more time");

    def random_input(required_datatype, min_value=None, max_value=None): """
This function generate random value for task.
:param datatype: required type of data
:param min_value: minimum required value for task
:param max_value: maximum required value for task
:return: random value
"""

    if required_datatype == int:
        generated_value = random.randint(min_value,max_value)

    elif required_datatype == float:
        generated_value = random.uniform(min_value,max_value)

    elif required_datatype == str:
        if min_value is None:
            min_value = 1

        if max_value is None:
            max_value = 50

        generated_value = ''.join(random.choices(
            string.ascii_letters + string.digits,
            k=random.randint(min_value, max_value)
        ))
    else:
        raise ValueError("Unused type of data for tasks, use: int, float, str")

    return generated_value

```

## Task1.py

```

import math import input_checker
def task1():
    """
    This function will calculate ln(x-1) using series decomposition,

```

```

        compare with math.ln(x-1).
"""
MAX_ITERATIONS = 500

# Input choose
choice = input_checker.input_checker("Input:"
                                     "\n1 -> for manual input"
                                     "\n2 -> for auto input", int, 1, 2)

if choice == 1:
    # Manual input
    while True:
        x = input_checker.input_checker("Input \'x\' greater than 1: ",
float, 1)
        if x == -1 or x == 1:
            print("\'x\' must not be lower than 1")
        else:
            break

    eps = input_checker.input_checker("Input \'eps\' from 0 to 1: ", float,
1e-10, 1)

else:
    # Auto input
    x = input_checker.random_input(float, 1, 2)
    eps = input_checker.random_input(float, 1e-10, 0.1)

# Calculating ln(x-1) with decomposition
y = x - 2 # Shift to center around 1
if abs(y) >= 1:
    raise ValueError("Series only converges for 1 < x < 3")

result = 0.0
for n in range(1, MAX_ITERATIONS + 1):
    term = (-1) ** (n + 1) * (y ** n) / n
    result += term
    if abs(term) < eps:
        break

# Value from math library
math_value = math.log(x-1)

# Display results
print("\nResults:")
print("|-----|")
print("|      x      | n |      F(x)      | Math F(x) | eps |")
print("|-----|-----|-----|-----|-----|")
print(f"| {x:7.4f} | {n + 1:3} | {result:0.3f} | {math_value:0.3f} | {eps:8.2e} |")
print("|-----|")

```

## Task2.py

```
import input_checker
def task2(): """ This function calculate count of numbers from 5 to 25
:return:count of numbers between 5 and 25 """
choice = input_checker.input_checker("Input:"
                                     "\n1 -> for manual input"
                                     "\n2 -> for auto input", int, 1, 2)

if choice == 1:
    count = 0
    while True:
        user_input = input_checker.input_checker("Input your number, input 0
to stop", (float,int))

        if user_input >= 5 or user_input <= 25:
            count += 1
        elif user_input == 0:
            print(f"Count of numebrs beetwen 5 and 25 is: {count}")
            break

    else:
        count = 0
        count_of_random_value = input_checker.random_input(int,10,20)

        while count_of_random_value:
            random_value = input_checker.random_input(int, -100, 100)

            if random_value >= 5 or random_value <= 25:
                count += 1
                count_of_random_value -= 1

        print("Auto stop")

        #Display results
        print(f"Count of numebrs beetwen 5 and 25 is: {count}")
```

## Task3.py

```
import input_checker from input_checker import random_input
def task3(): """ This function count of spaces and apos in text :return:
count of spaces and apos """
# Input choose
choice = input_checker.input_checker("Input:"
                                     "\n1 -> for manual input"
                                     "\n2 -> for auto input", int, 1, 2)

if choice == 1:
    count_of_space = 0
    count_of_apo = 0
    user_inpur = input_checker.input_checker("Input your text",str)
```

```

for symbol in user_inpur:

    if symbol == " ":
        count_of_space += 1

    elif symbol == "'":
        count_of_apo += 1

print(f"Count of space: {count_of_space}"
      f"\nCount of apo: {count_of_apo}")

else:
    count_of_space = 0
    count_of_apo = 0
    random_text = random_input(str,10,20)
    for symbol in random_text:

        if symbol == " ":
            count_of_space += 1

        elif symbol == "'":
            count_of_apo += 1

    print(f"Count of space: {count_of_space}"
          f"\nCount of apo: {count_of_apo}")

```

#### Task4.py

```

import input_checker
def process_text(text): # Split text by words words = text.split()
    # a) count words with len < 6
    short_words_count = sum(1 for word in words if len(word) < 6)
    print(f"Count of words with len < 6: {short_words_count}")

    # b) find the shortest word with "w" in the end
    words_ending_with_w = [word for word in words if word.endswith('w')]
    if words_ending_with_w:
        shortest_word = min(words_ending_with_w, key=len)
        print(f"The shortest word with 'w' in the end: {shortest_word}")
    else:
        print("There are no word with 'w' in the end.")

    # c) print sorted words by length
    sorted_words = sorted(words, key=len)
    print("Sorted words by their length:")
    for word in sorted_words:
        print(word)

def task4(): """ This function count words with length lower than 6,
find the shortest word with last letter = w, sort words by their length
:return: words which len < 6, shortest word with "w" in the end, sorted words
by length """

```

```

# Input choose
choice = input_checker.input_checker("Input:"
                                     "\n1 -> for finished text"
                                     "\n2 -> for random input", int, 1, 2)

text = "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."

random_generated_text = input_checker.random_input(str,10,50)

if choice == 1:
    process_text(text)
else:
    process_text(random_generated_text)

```

## Task5.py

```

import input_checker

def process_list(numbers: list): # Генератор для нахождения индексов
нелулевых элементов
def non_zero_indices():
for i, num in enumerate(numbers):
if num != 0: yield i

# Находим индекс максимального значения
max_value = max(numbers)
max_index = numbers.index(max_value)
print(f"Index of the greatest number: {max_index}")

# Используем генератор для получения первых двух ненулевых индексов
nz_indices = non_zero_indices()
first_non_zero_index = next(nz_indices, None)
second_non_zero_index = next(nz_indices, None)

# Вычисляем произведение между двумя первыми ненулевыми элементами
if first_non_zero_index is not None and second_non_zero_index is not None:
    product = 1
    for num in numbers[first_non_zero_index + 1:second_non_zero_index]:
        product *= num
    print(f"Product between two non-zero numbers: {product}")
else:
    print("There are not enough non-zero numbers.")

def task5(): """ This function find index of the greatest number,
product of two non-zero numbers :return: """
# Input choose
choice = input_checker.input_checker("Input:"
                                     "\n1 -> for manual text"
                                     "\n2 -> for random input", int, 1, 2)

listik = []
if choice == 1:

```

```

        count_user_input = input_checker.input_checker("Input count of numbers
for list to check",int,1)

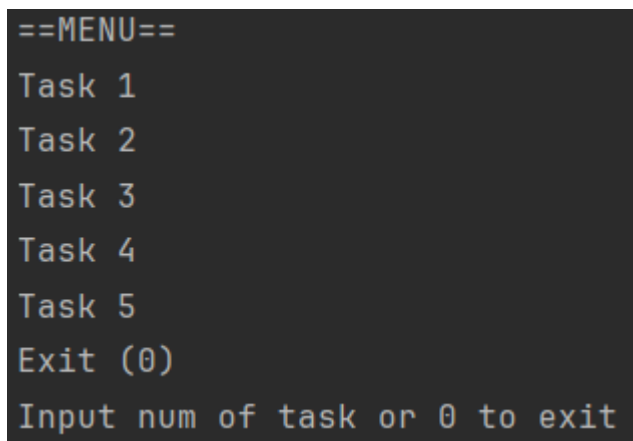
        while count_user_input:
            user_input = input_checker.input_checker("Input num for the
list",int)
            listik.append(user_input)
            count_user_input -= 1

        process_list(listik)

    else:
        random_count_of_numbers = input_checker.random_input(int,5,15)
        while random_count_of_numbers:
            random_input = input_checker.random_input(int,-50,50)
            listik.append(random_input)
            random_count_of_numbers -= 1

        process_list(listik)

```



```

==MENU==
Task 1
Task 2
Task 3
Task 4
Task 5
Exit (0)
Input num of task or 0 to exit

```

Рисунок 1 - Главное меню программы



```

Input num of task or 0 to exit 1
Input:
1 -> for manual input
2 -> for auto input1
Input 'x' greater than 1: 2
Input 'eps' from 0 to 1: 0.3

Results:
|-----|
|      x      | n |      F(x)      | Math F(x) |      eps      |
|-----|-----|-----|-----|-----|
|  2.0000  |  2 |  0.000  |  0.000  |  3.00e-01  |
|-----|

```

Рисунок 2 - Выполнение 1 задания

```

Input num of task or 0 to exit 2
Input:
1 -> for manual input
2 -> for auto input2
Auto stop
Count of numebrs beetwen 5 and 25 is: 14

```

Рисунок 3 - Выполнение 2 задания

```

Input num of task or 0 to exit 3
Input:
1 -> for manual input
2 -> for auto input1
Input your textasd ' ' asd
Count of space: 3
Count of apo: 2

```

Рисунок 4 - выполнение 3 задания

```
Input num of task or 0 to exit 4
Input:
1 -> for finished text
2 -> for random input1
Count of words with len < 6: 41
There are no word with 'w' in the end.
Sorted words by their length:
a
a
So
in
as
as
of
be
of
```

Рисунок 5 - выполнение 4 задания

```
Input num of task or 0 to exit 5
Input:
1 -> for manual text
2 -> for random input2
Index of the greatest number: 3
Product between two non-zero numbers: 1
```

Рисунок 6 - выполнение 5 задания

```
==MENU==  
Task 1  
Task 2  
Task 3  
Task 4  
Task 5  
Exit (0)  
Input num of task or 0 to exit 0  
Exit  
  
Process finished with exit code 0
```

Рисунок 7 - Завершение работы программы