

Test1 – Set1

Enter your personal details:

Name and surname:

Student ID number:

Signature:

Turn off your mobile phone and do not use any aids during the test. If you finish the test before deadline, you can leave the computer lab.

Complete the following tasks. You have 70 minutes to complete the test. Save the results (created functions) to the files with names given in brackets. Upload the files to the Moodle platform. ATTENTION. Check that the names of the created files and the names of the created functions match the contents of the task.

(F1.py) (3 pts) Define a function `f1(n1,n2,n3)` that returns true when all three numbers `n1,n2,n3` are different. Example:

`f1(4,8,5) => True`

`f1(2,9,2) => False`

(F2.py) (4 pts) Define a function `f2()` that returns the text "yes" or "no" randomly. Example:

`f2() => "yes"`

`f2() => "no"`

(F3.py) (4 pts) Define a function `f3(n)` that returns the sum of digits of `n`. Example:

`f3(39126) => 21`

`f3(27) => 9`

(F4.py) (4 pts) Define a function `f4(n1,n2,n3)` that returns the result of subtraction between the largest and smallest of the numbers `n1,n2,n3`. Example:

`f4(7,4,9) => 5`

`f4(3,2,1) => 2`

(F5.py) (4 pts) Define a function `f5(n)` that returns sequential numbers 1 through `n` as a string. Example:

`f5(11) => "1234567891011"`

`f5(4) => "1234"`

(F6.py) (5 pts) Define a function `f6(x,n)` that returns `n`-th odd term of an arithmetic sequence with an initial term of 1 and difference of `x`. Example:

`f6(2,8) => 15`

`f6(3,4) => 19`

(F7.py) (6 pts) Define a function `f7(n)` that returns the `n`-th prime number. A prime number is a natural number greater than 1, divisible by 1 and by that number. Example:

`f7(1) => 2`

`f7(5) => 11`

Test1 – Zestaw2

Wpisz swoje dane personalne:

Imię i nazwisko:

Nr legitymacji:

Podpis:

Wyłącz telefon komórkowy oraz nie korzystaj z żadnych pomocy podczas kolokwium. Jeśli zakończysz kolokwium przed czasem, możesz opuścić salę komputerową.

Wykonaj poniższe zadania. Na ich wykonanie masz 70 minut. Rezultaty (utworzone funkcje) zapisz do plików o nazwach podanych w nawiasach. Pliki prześlij na platformę Moodle. UWAGA. Sprawdź, czy nazwy utworzonych plików oraz nazwy utworzonych funkcji są zgodne z treścią zadania.

(F1.py) (3 pkt) Zdefiniuj funkcję $f1(n1, n2, n3)$, która zwraca prawdę, gdy wszystkie trzy liczby $n1, n2, n3$ są różne. Przykład:

$f1(4, 8, 5) \Rightarrow \text{True}$

$f1(2, 9, 2) \Rightarrow \text{False}$

(F2.py) (4 pkt) Zdefiniuj funkcję $f2()$, która zwraca losowo tekst „tak” lub „nie”. Przykład:

$f2() \Rightarrow \text{"tak"}$

$f2() \Rightarrow \text{"nie"}$

(F3.py) (4 pkt) Zdefiniuj funkcję $f3(n)$, która zwraca sumę cyfr w liczbie n . Przykład:

$f3(39126) \Rightarrow 21$

$f3(27) \Rightarrow 9$

(F4.py) (4 pkt) Zdefiniuj funkcję $f4(n1, n2, n3)$, która zwraca różnicę pomiędzy największą i najmniejszą z liczb $n1, n2, n3$. Przykład:

$f4(7, 4, 9) \Rightarrow 5$

$f4(3, 2, 1) \Rightarrow 2$

(F5.py) (4 pkt) Zdefiniuj funkcję $f5(n)$, która zwraca kolejne liczby od 1 do n , jako ciąg znakowy.

Przykład:

$f5(11) \Rightarrow \text{"1234567891011"}$

$f5(4) \Rightarrow \text{"1234"}$

(F6.py) (5 pkt) Zdefiniuj funkcję $f6(x, n)$, która zwraca n -ty nieparzysty wyraz ciągu arytmetycznego o wyrazie początkowym równym 1 i różnicy ciągu x . Przykład:

$f6(2, 8) \Rightarrow 15$

$f6(3, 4) \Rightarrow 19$

(F7.py) (6 pkt) Zdefiniuj funkcję $f7(n)$, która zwraca n -tą liczbę pierwszą. Liczba pierwsza to liczba naturalna większa od 1, podzielna przez 1 oraz przez tę liczbę. Przykład:

$f7(1) \Rightarrow 2$

$f7(5) \Rightarrow 11$

Test1 – Zestaw3

Wpisz swoje dane personalne:

Imię i nazwisko:

Nr legitymacji:

Podpis:

Wyłącz telefon komórkowy oraz nie korzystaj z żadnych pomocy podczas kolokwium. Jeśli zakończysz kolokwium przed czasem, możesz opuścić salę komputerową.

Wykonaj poniższe zadania. Na ich wykonanie masz 70 minut. Rezultaty (utworzone funkcje) zapisz do plików o nazwach podanych w nawiasach. Pliki prześlij na platformę Moodle. UWAGA. Sprawdź, czy nazwy utworzonych plików oraz nazwy utworzonych funkcji są zgodne z treścią zadania.

(F1.py) (3 pkt) Zdefiniuj funkcję $f1(n1, n2, n3)$, która zwraca prawdę, jeśli choć jedna z liczb $n1, n2, n3$ jest ujemna. Przykład:

$f1(11, 6, -4) \Rightarrow \text{True}$

$f1(5, 4, 14) \Rightarrow \text{False}$

(F2.py) (4 pkt) Zdefiniuj funkcję $f2(n)$, która zwraca łańcuch znakowy składający się z n gwiazdek, rozdzielonych znakiem ukośnika. Przykład:

$f2(4) \Rightarrow "**/*/*/*/*"$

$f2(1) \Rightarrow "**"$

(F3.py) (4 pkt) Zdefiniuj funkcję $f3(x, y)$, która zwraca, ile jest ujemnych liczb parzystych w przedziale $\langle x, y \rangle$. Przykład:

$f3(-7, 8) \Rightarrow 3$

$f3(-1, 11) \Rightarrow 0$

(F4.py) (4 pkt) Zdefiniuj funkcję $f4()$, która zwraca losowo liczbę 3, 7 lub 11. Przykład:

$f4() \Rightarrow 7$

$f4() \Rightarrow 3$

(F5.py) (4 pkt) Zdefiniuj funkcję $f5(n)$, która zwraca, ile cyfr parzystych zawiera liczba n . Przykład:

$f5(314569) \Rightarrow 2$

$f5(2274) \Rightarrow 3$

(F6.py) (5 pkt) Zdefiniuj funkcję $f6(x, n)$, która zwraca n -ty nieparzysty wyraz ciągu arytmetycznego o wyrazie początkowym równym 1 i różnicy ciągu x . Przykład:

$f6(2, 8) \Rightarrow 15$

$f6(3, 4) \Rightarrow 19$

(F7.py) (6 pkt) Zdefiniuj funkcję $f7(n)$, która zwraca n -ty wyraz ciągu Fibonacciego. Ciąg jest zdefiniowany następująco: pierwszy wyraz ciągu jest równy 0, drugi wyraz jest równy 1. Każdy następny wyraz stanowi sumę dwóch poprzednich. Przykład:

$f7(5) \Rightarrow 3$

$f7(9) \Rightarrow 21$

Test1 – Set4

Enter your personal details:

Name and surname:

Student ID number:

Signature:

Turn off your mobile phone and do not use any aids during the test. If you finish the test before deadline, you can leave the computer lab.

Complete the following tasks. You have 70 minutes to complete the test. Save the results (created functions) to the files with names given in brackets. Upload the files to the Moodle platform. ATTENTION. Check that the names of the created files and the names of the created functions match the contents of the task.

(F1.py) (3 pts) Define a function `f1(n1,n2,n3)` that returns true when all three numbers `n1,n2,n3` are different. Example:

`f1(4,8,5) => True`

`f1(2,9,2) => False`

(F2.py) (4 pts) Define a function `f2()` that returns the text "yes" or "no" randomly. Example:

`f2() => "yes"`

`f2() => "no"`

(F3.py) (4 pts) Define a function `f3(n)` that returns the sum of digits of `n`. Example:

`f3(39126) => 21`

`f3(27) => 9`

(F4.py) (4 pts) Define a function `f4(n1,n2,n3)` that returns the result of subtraction between the largest and smallest of the numbers `n1,n2,n3`. Example:

`f4(7,4,9) => 5`

`f4(3,2,1) => 2`

(F5.py) (4 pts) Define a function `f5(n)` that returns sequential numbers 1 through `n` as a string.

Example:

`f5(11) => "1234567891011"`

`f5(4) => "1234"`

(F6.py) (5 pts) Define a function `f6(x,n)` that returns `n`-th odd term of an arithmetic sequence with an initial term of 1 and difference of `x`. Example:

`f6(2,8) => 15`

`f6(3,4) => 19`

(F7.py) (6 pts) Define a function `f7(n)` that returns the `n`-th prime number. A prime number is a natural number greater than 1, divisible by 1 and by that number. Example:

`f7(1) => 2`

`f7(5) => 11`

Test1 – Zestaw 5

Wpisz swoje dane personalne:

Imię i nazwisko:

Nr legitymacji:

Podpis:

Wyłącz telefon komórkowy oraz nie korzystaj z żadnych pomocy podczas kolokwium. Jeśli zakończysz kolokwium przed czasem, możesz opuścić salę komputerową.

Wykonaj poniższe zadania. Na ich wykonanie masz 70 minut. Rezultaty (utworzone funkcje) zapisz do plików o nazwach podanych w nawiasach. Pliki prześlij na platformę Moodle. UWAGA. Sprawdź, czy nazwy utworzonych plików oraz nazwy utworzonych funkcji są zgodne z treścią zadania.

(F1.py) (3 pkt) Zdefiniuj funkcję $f1(n1, n2, n3)$, która zwraca prawdę, jeśli choć jedna z liczb $n1, n2, n3$ jest ujemna. Przykład:

$f1(11, 6, -4) \Rightarrow \text{True}$

$f1(5, 4, 14) \Rightarrow \text{False}$

(F2.py) (4 pkt) Zdefiniuj funkcję $f2(n)$, która zwraca łańcuch znakowy składający się z n gwiazdek, rozdzielonych znakiem ukośnika. Przykład:

$f2(4) \Rightarrow "**/*/*/*/*"$

$f2(1) \Rightarrow "**"$

(F3.py) (4 pkt) Zdefiniuj funkcję $f3(x, y)$, która zwraca, ile jest ujemnych liczb parzystych w przedziale $\langle x, y \rangle$. Przykład:

$f3(-7, 8) \Rightarrow 3$

$f3(-1, 11) \Rightarrow 0$

(F4.py) (4 pkt) Zdefiniuj funkcję $f4()$, która zwraca losowo liczbę 3, 7 lub 11. Przykład:

$f4() \Rightarrow 7$

$f4() \Rightarrow 3$

(F5.py) (4 pkt) Zdefiniuj funkcję $f5(n)$, która zwraca, ile cyfr parzystych zawiera liczba n . Przykład:

$f5(314569) \Rightarrow 2$

$f5(2274) \Rightarrow 3$

(F6.py) (5 pkt) Zdefiniuj funkcję $f6(x, n)$, która zwraca n -ty nieparzysty wyraz ciągu arytmetycznego o wyrazie początkowym równym 1 i różnicy ciągu x . Przykład:

$f6(2, 8) \Rightarrow 15$

$f6(3, 4) \Rightarrow 19$

(F7.py) (6 pkt) Zdefiniuj funkcję $f7(n)$, która zwraca n -ty wyraz ciągu Fibonacciego. Ciąg jest zdefiniowany następująco: pierwszy wyraz ciągu jest równy 0, drugi wyraz jest równy 1. Każdy następny wyraz stanowi sumę dwóch poprzednich. Przykład:

$f7(5) \Rightarrow 3$

$f7(9) \Rightarrow 21$