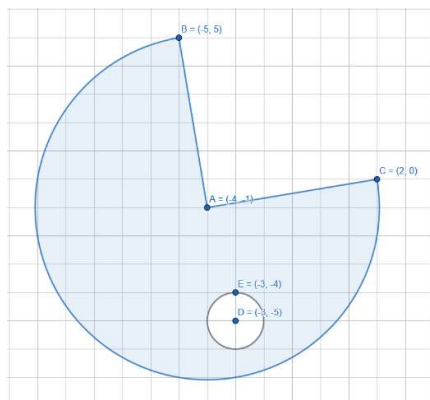


Вариант 1	група за семинарни упражнения	
Име, фамилия		ф.н.

Задача 1.



А) Напишете логически израз, който се оценява като истина, ако точка с координати (x, y) принадлежи на оцветената област, и като лъжа в противен случай.

Б) Напишете функция, която по две валидни точки от шахматна дъска проверява дали може да се достигне от първата до втората точка с един ход на топ, цар или царица.

Задача 2.

Напишете програма, която въвежда от стандартния вход цели числа a_i и b_i до въвеждане на числото 0 и намира сумата на рационалните числа $\frac{a_i}{b_i}$. Резултатът да се изведе като смесено число.

Вход: 1 2 2 3 3 4 1 0

Изход: 2 и 5/12

Задача 3.

Генетичният алгоритъм е вдъхновен от теорията за еволюцията на Дарвин. Алгоритъмът стартира с множество от потенциални решения (индивиди), които се генерират случайно. Наричаме ги начална популация.

Размерът е цяло неотрицателно число *pop_size*.

Следващо поколение се получава по следния начин. Най-добрите k , $k < pop_size$, индивида от текущата популация се запазват директно в следващото поколение. Останалите $pop_size - k$ индивида се получават след кръстосване на произволно избрани родителски индивиди от текущата популация. Накрая всеки нов индивид преминава през мутация.

Алгоритъмът разчита на това, че всяко следващо поколение е по-добро от предходното. Приключва, когато сред текущата популация намери решението, което се търси.

Напишете програма, която използва генетичен алгоритъм, за да познае цяло число без знак, въведено от стандартния вход. За целта реализирайте следните функции:

А) Функция, която получава като аргумент цяло неотрицателно число *pop_size* и генерира *pop_size* на брой цели случайни числа, които представляват началната популация на генетичния алгоритъм.

Б) Функция, която по две цели числа без знак (unsigned int) намира броя битове, в които двете числа се различават (разстояние по Хеминг).

В) Функция, която по подаден масив от потенциални решения, избира първите k най-добри. k е цяло число без знак. При оценяване на потенциалните решения се използва разстоянието по Хеминг. Колкото по-малко е разстоянието по Хеминг между потенциално решение и търсеното число, толкова по-добро е решението.

Г) Функция, която получава две цели числа без знак. Те кодират хромозомите на два родителски индивида. Функцията реализира двуточково кръстосване в случайно генерирани различни позиции в интервала $[1; 8 \cdot \text{sizeof}(\text{unsigned int})]$. В резултат се получават два нови индивида.

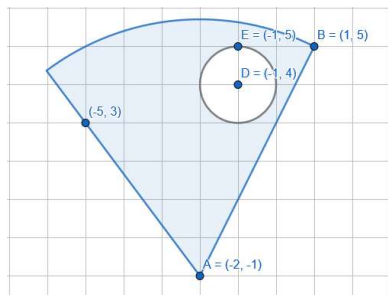
Д) Функция, която получава като аргумент цяло число без знак (unsigned int), което е потенциално решение или индивид, и обръща един бит на случайно генерирана позиция. Функцията реализира генетичната операция мутация.

Максималният брой индивиди в популацията, който допускаме за конкретната задача, е 100.

Решение от текущата популация съвпада с търсеното число, ако разстоянието по Хеминг между двете е 0.

Вариант 2	група за семинарни упражнения		
Име, фамилия		ф.н.	

Задача 1.



А) Напишете логически израз, който се оценява като истина, ако точка с координати (x, y) принадлежи на оцветената област, и като лъжа в противен случай.

Б) Напишете функция, която по две валидни точки от шахматна дъска проверява дали може да се достигне от първата до втората точка с един ход на кон, офицер или топ.

Задача 2.

Напишете програма, която въвежда от стандартния вход три цели числа a , $b > 0$, $k > 0$ и намира $\sum_{i=1}^k \frac{a+i}{b+i}$. Резултатът да се изведе като смесено число.

Вход: 3 2 4

Изход: 4 и 19/20

Задача 3.

Генетичният алгоритъм е вдъхновен от теорията за еволюцията на Дарвин. Алгоритъмът стартира с множество от потенциални решения (индивиди), които се генерират случайно. Наричаме ги начална популация. Размерът е цяло неотрицателно число *pop_size*.

Следващо поколение се получава по следния начин. Най-добрите k , $k < pop_size$, индивида от текущата популация се запазват директно в следващото поколение. Останалите $pop_size - k$ индивида се получават след кръстосване на произволно избрани родителски индивиди от текущата популация. Накрая всеки нов индивид преминава през мутация с вероятност *pt*.

Алгоритъмът разчита на това, че всяко следващо поколение е по-добро от предходното. Приключва след определен брой повторения, зададен с параметъра *max_iter*. Резултатът е най-доброто решение в последното поколение.

Напишете програма, която използва генетичен алгоритъм, за да познае число, което е въведено от стандартния вход. За целта реализирайте следните функции:

А) Функция, която получава като аргумент цяло неотрицателно число *pop_size* и генерира *pop_size* на брой цели случайни числа, които представляват началната популация на генетичния алгоритъм.

Б) Функция, която по две цели числа без знак (unsigned int) намира броя битовете, в които двете числа се различават (разстояние по Хеминг).

В) Функция, която по подаден масив от потенциални решения, избира първите k най-добри. k е цяло число без знак. При оценяване на потенциалните решения се използва разстоянието по Хеминг. Колкото по-малко е разстоянието по Хеминг между потенциално решение и търсеното число, толкова по-добро е решението.

Г) Функция, която получава две цели числа без знак. Те кодират хромозомите на два родителски индивида. Функцията реализира двуточково кръстосване в случайно генерирани различни позиции в интервала $[1; 8 \cdot \text{sizeof}(\text{unsigned int})]$. В резултат се получават два нови индивида.

Д) Функция, която получава като аргументи потенциално решение (unsigned int) и вероятност за мутация *pt*, $pt \in [0; 100)$. Функцията генерира случайно число между 0 и 99 и ако неговата стойност е по-малка от *pt*, обръща бита на случайно генерирана позиция, иначе върху подаденото решение не се променя.

Максималният брой индивиди в популацията, който допускаме за конкретната задача, е 100.

Решение от текущата популация съвпада с търсеното число, ако разстоянието по Хеминг между двете е 0.

Допълнителна задача

Редицата "look-and-say" се генерира по следния начин.

Стартираме с произволно цяло число без знак. Следващият елемент на редицата се получава от предходния, като прочитаме неговите цифри. Еднаквите последователни цифри се преброяват и се отбелязва този брой.

Например:

Начало: 2

2: изчитаме *една двойка*; записваме 1 2

1 2: изчитаме *една единица, една двойка*; записваме 1 1 1 2

1 1 1 2: изчитаме *три единици, една двойка*; записваме 3 1 1 2

3 1 1 2: изчитаме *една тройка, две единици, една двойка*; записваме 1 3 2 1 1 2

Да се напише програма, която извежда първите k на брой елемента на "look-and-say" редица, чието начало се въвежда от стандартния вход. k също се въвежда от стандартния вход.