Національний Технічний Університет України

Київський Політехнічний Інститут імені Ігоря Сікорського

Інститут Прикладного Системного Аналізу

Кафедра Системного Проектування

ЛАБОРАТОРНА РОБОТА №5

З дисципліни: «Теорія прийняття рішень»

На тему

«ПРИЙНЯТТЯ РІШЕНЬ ЗА ДОПОМОГОЮ МЕТОДІВ ГОЛОСУВАННЯ»

Виконала:

Студентка групи ДА-81

Желєзнова В.С.

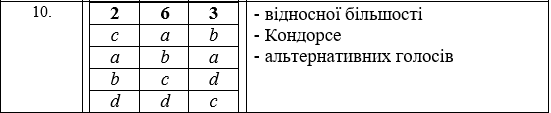
Варіант №10

Київ – 2021

Мета роботи:

Ознайомитись з методами прийняття рішень за допомогою методів голосування.

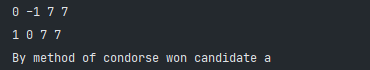
Варіант:

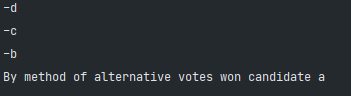


Результати роботи програми:

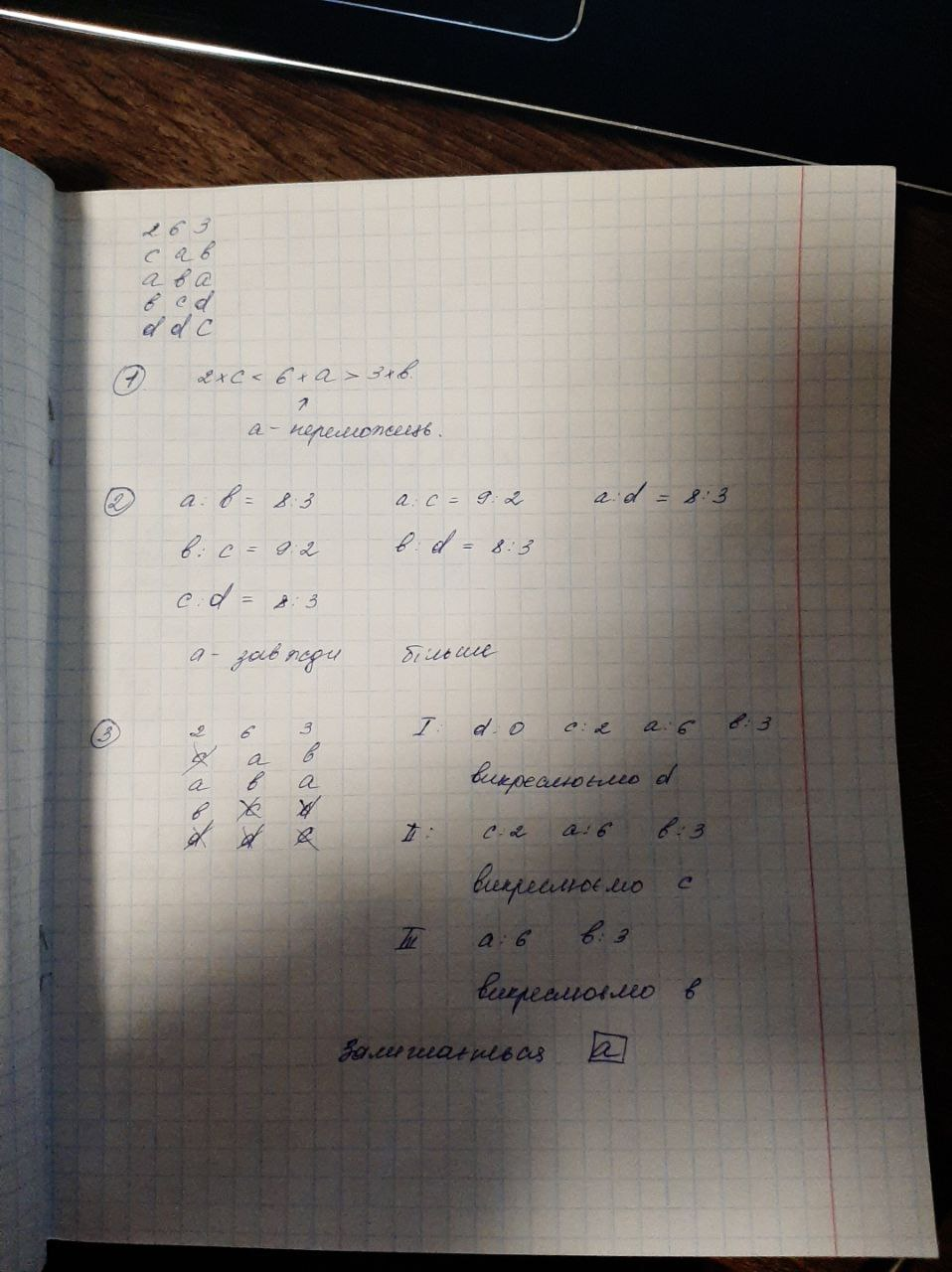
* Відносної більшості



* Кондорсе  
  
* Альтернативних голосів



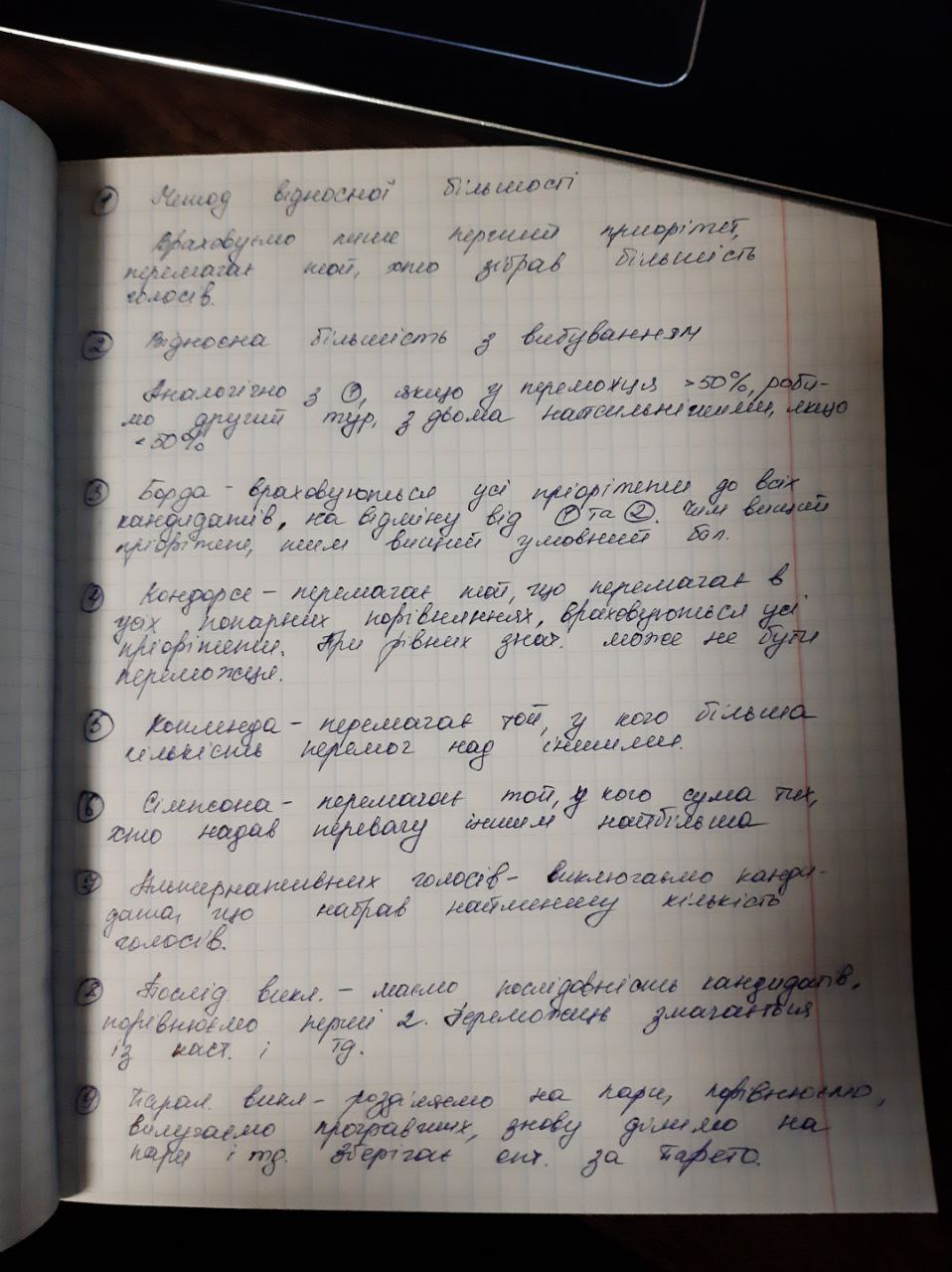
Аналітичний розв’язок:



Висновок:

Загалом нам трапився такий набір даних що у другому стовпчику більше всього виборців, навіть суми двох інших, що робить варіант а майже гарантованим переможцем при більшій частині типів виборів.

Контрольні питання:



Лістинг:

public class Election {  
  
 static int[] *sameVotesNumber* = new int[]{2,6,3};  
 static char[][] *votesGroupes* = new char[][]{{'c','a','b','d'},{'a','b','c','d'},{'b','a','d','c'}};  
  
 static char vidnosnoiBilshosti(){  
 int[] participantVotes = new int[4];  
 //враховуємо тільки перший пріоритет  
 for (int i = 0; i < *sameVotesNumber*.length; i++) {  
 participantVotes[(int)*votesGroupes*[i][0]-(int)'a']+=*sameVotesNumber*[i];  
 }  
 int maxIndex = 0;  
 //знаходимо кандидата з найбільшою кількістю виборців  
 for (int i = 0; i < participantVotes.length; i++) {  
 if(participantVotes[i]>participantVotes[maxIndex]) maxIndex = i;  
 }  
 return (char)((int)'a'+maxIndex);  
 }  
 static char Condorse(){  
 //проходимось по набору кандидатів  
 for (int i = 0; i < *votesGroupes*[2].length; i++) {  
 int[] candidateVsOthers = new int[4];  
 //проходимось по всім стовпчикам з пріоритетами  
 for (int j = 0; j < *votesGroupes*.length; j++) {  
 // в кожному стовпчику дивимося чи досягли ми кандидата по списку і якщо вище кандидата то йде в мінус до загального  
 boolean found = false;  
 for (int k = 0; k < *votesGroupes*[j].length; k++) {  
 if(*votesGroupes*[j][k]==*votesGroupes*[2][i]){  
 found = true;  
 }else if(found){  
 candidateVsOthers[(int)*votesGroupes*[j][k]-(int)'a']+=*sameVotesNumber*[j];  
 }else {  
 candidateVsOthers[(int)*votesGroupes*[j][k]-(int)'a']-=*sameVotesNumber*[j];  
 }  
 }  
 }  
 for (int num: candidateVsOthers) {  
 System.*out*.print(num+"\t");  
 }  
 boolean foundBest = true;  
 for (int j = 0; j < candidateVsOthers.length; j++) {  
 if(candidateVsOthers[j] < 0){  
 //якщо ми бачимо що наш кандидат програє певному іншому кандидату то на наступній ітерації ми перевіряємо його  
 i = j - 1;  
 foundBest = false;  
 }  
 }  
 System.*out*.println();  
 if(foundBest){  
 return *votesGroupes*[2][i];  
 }  
  
 }  
  
  
  
 return '-';  
 }  
 static char alternativeVotes(){  
 int[] participantVotes = new int[4];  
 for (int i = 0; i < 4; i++) {  
 //рахуємо кількість голоів враховуючи що дехто міг вибути  
 for (int j = 0; j < *sameVotesNumber*.length; j++) {  
 int firstVote = 0;  
 for(int k = 0; participantVotes[(int)(*votesGroupes*[j][k])-(int)'a']==-1;k++)  
 firstVote++;  
 participantVotes[(int)*votesGroupes*[j][firstVote]-(int)'a']+=*sameVotesNumber*[j];  
 }  
 if(i < 3){  
 //вилучаємо найгіршого кандидата  
 int minIndex = 0;  
 for (int j = 0; j < participantVotes.length; j++) {  
 if(participantVotes[j]!=-1 && participantVotes[j]<participantVotes[minIndex]) minIndex = j;  
 }  
 participantVotes[minIndex]=-1;  
 System.*out*.println("-"+(char)((int)'a'+(int)minIndex));  
 }  
 }  
 for (int i = 0; i < 4; i++) {  
 if(participantVotes[i]!=-1) return (char)((int)'a'+ i);  
 }  
 return '-';  
 }  
  
 public static void main(String[] args) {  
 System.*out*.println("By method of vidnosnoi bilshisti won candidate "+ *vidnosnoiBilshosti*() );  
 System.*out*.println("By method of condorse won candidate "+ *Condorse*() );  
 System.*out*.println("By method of alternative votes won candidate "+ *alternativeVotes*() );  
 }  
}