

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В. Г. ШУХОВА»
(БГТУ им. В.Г. Шухова)**

Кафедра программного обеспечения вычислительной техники и автоматизированных
систем

Лабораторная работа № 2.2

по дисциплине: Дискретная математика
тема: «Задачи выбора»

Выполнил: ст. группы ПВ-202

Аладиб язан

Проверил:

Рязанов Юрий Дмитриевич

Бондаренко Татьяна

Владимировна

Лабораторная работа № 2.2

«Задачи выбора»

Цель работы:

приобрести практические навыки в использовании алгоритмов порождения комбинаторных объектов при проектировании алгоритмов решения задач выбора

Задания к работе:

1. Ознакомиться с задачей (см. варианты заданий).
2. Определить класс комбинаторных объектов, содержащих решение задачи (траекторию задачи).
3. Определить, что в задаче является функционалом и способ его вычисления.
4. Определить способ распознавания решения по значению функционала.
5. Реализовать алгоритм решения задачи.
6. Подготовить тестовые данные и решить задачу.

Задание варианта №2:

Числа из заданного девятиэлементного множества разместить в —числовом колесе (рис.2) так, чтобы одно число было в центре колеса, остальные — у концов каждого диаметра, и чтобы суммы чисел каждого ряда были бы одинаковыми.

1)

Числа из заданного девятиэлементного множества разместить в —числовом колесе (рис.2) так, чтобы одно число было в центре колеса, остальные — у концов каждого диаметра, и чтобы суммы чисел каждого ряда были бы одинаковыми.

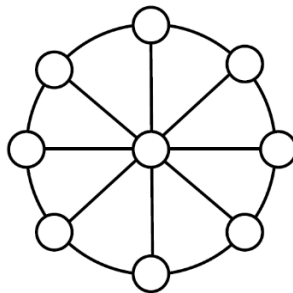


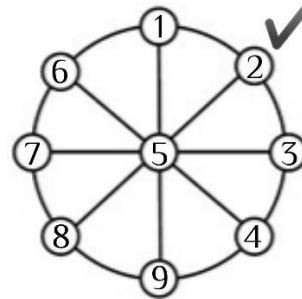
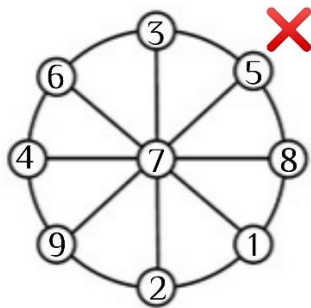
Рис.2. Числовое колесо

2)

количество комбинаций, в которых сумма рядов в —числовом колесе одинаковыми.

3)

Поместите 9 чисел в колесо так, чтобы три числа на прямой давали одно и то же число.



4)

Если сумма чисел во всех строках равна, результат будет правильным (true), а если есть разница, результат будет ложным (false).

5)

```
#include <stdio.h>

#define WHEEL_SIZE 9

// Function to check if the sums of numbers in each diameter of the wheel are equal.
int check_sums(int arr[]) {
    int sum_d1 = arr[0] + arr[4] + arr[8];
    int sum_d2 = arr[1] + arr[5] + arr[8];
    int sum_d3 = arr[2] + arr[6] + arr[8];
    int sum_d4 = arr[3] + arr[7] + arr[8];

    return (sum_d1 == sum_d2) && (sum_d2 == sum_d3) && (sum_d3 == sum_d4);
}

void print_wheel(int arr[], int *valid_count, int *invalid_count) {
    if (check_sums(arr)) {
        (*valid_count)++;
    }
}
```

```

        printf("[ ");
        for (int i = 0; i < WHEEL_SIZE; i++) {
            printf("%d ", arr[i]);
        }
        printf("] ----> Equal sums\n");
    } else {
        (*invalid_count)++;
    }
}

// Function to swap two numbers.
void swap(int *a, int *b) {
    int temp = *a;
    *a = *b;
    *b = temp;
}

// Function to generate all permutations of the array and check each one.
void generate_permutations(int *arr, int start, int end, int *total_count, int
*valid_count, int *invalid_count) {
    if (start == end) {
        (*total_count)++;
        print_wheel(arr, valid_count, invalid_count);
        return;
    }

    for (int i = start; i <= end; i++) {
        swap(&arr[i], &arr[start]);
        generate_permutations(arr, start + 1, end, total_count, valid_count,
invalid_count);
        swap(&arr[i], &arr[start]);
    }
}

int main() {
    int arr[WHEEL_SIZE];
    int total_count = 0, valid_count = 0, invalid_count = 0;

    printf("Enter the numbers for the wheel (9 numbers):\n");

    for (int i = 0; i < WHEEL_SIZE; i++) {
        if (scanf("%d", &arr[i]) != 1) {
            printf("Invalid input! Please enter numbers only.\n");
            return 1;
        }
    }
}

```

```

    }

    printf("Generating permutations and checking for equal sums across
diameters...\n");

    generate_permutations(arr, 0, WHEEL_SIZE - 1, &total_count, &valid_count,
&invalid_count);

    printf("\nSummary:\n");
    printf("Total arrangements: %d\n", total_count);
    printf("Valid arrangements (equal sums): %d\n", valid_count);
    printf("Invalid arrangements (unequal sums): %d\n", invalid_count);

    return 0;
}

```

6)

Количество рядов = $9! = 362880$.

числа колеса (входные числа)	Количество рядов с одинаковой суммой	Количество рядов с неравной суммой
1,2,3,4,5,6,7,8,9	1152	361728
1,3,5,9,11,13,15,17,19	384	362496
1,3,5,7,9,10,11,12,13	0	362880