

```
#include <stdio.h>
#include <locale.h>

void shiftArrayRight(int* array, int size) {
    int temp = array[size - 1];
    for (int i = size - 1; i > 0; i--) {
        array[i] = array[i - 1];
    }
    array[0] = temp;
}

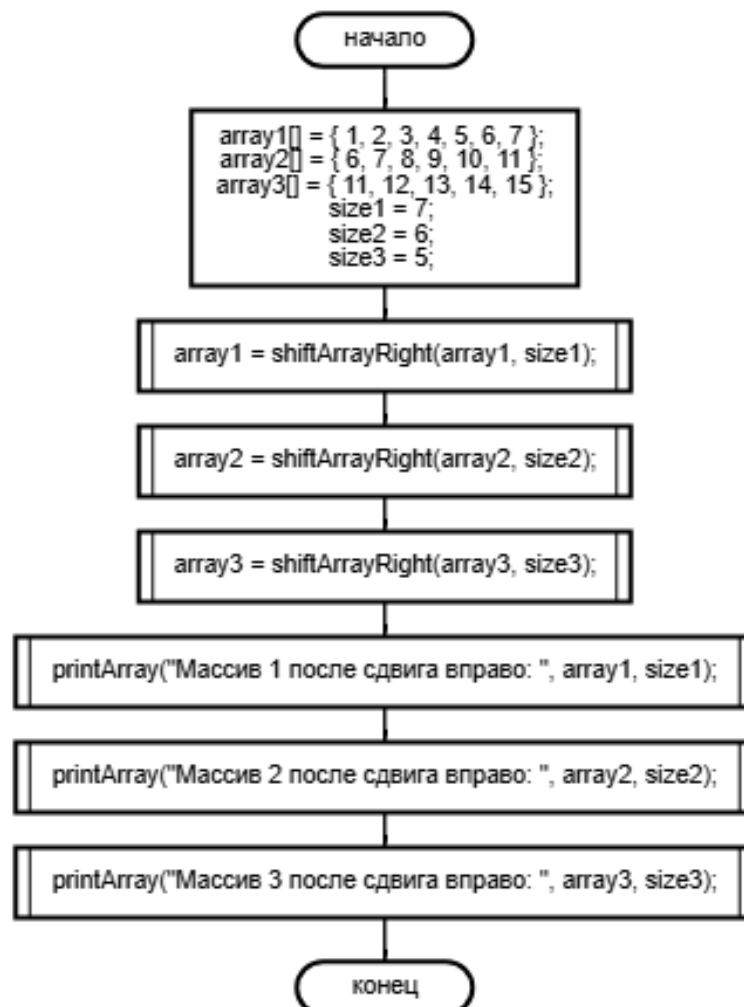
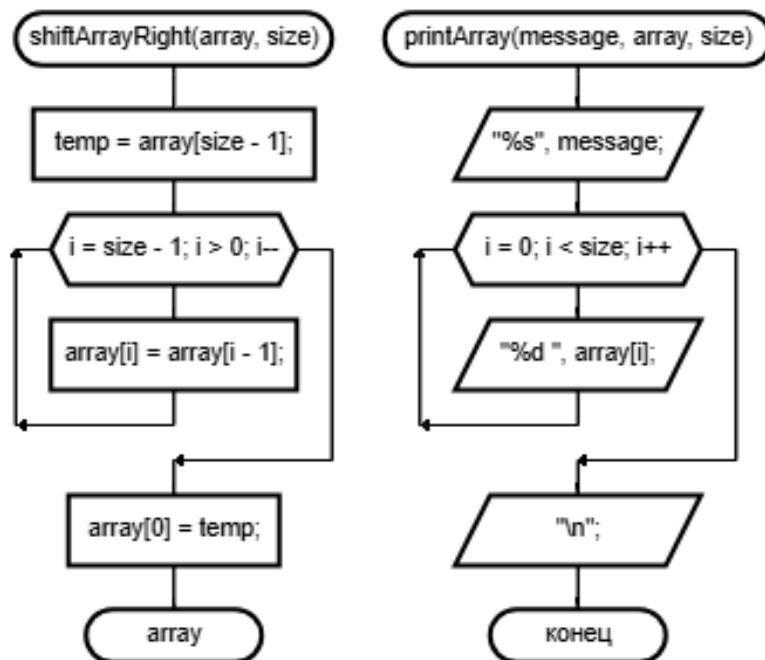
void printArray(const char* message, int* array, int size) {
    printf("%s", message);
    for (int i = 0; i < size; i++) {
        printf("%d ", array[i]);
    }
    printf("\n");
}

int main() {
    setlocale(LC_ALL, "RU_ru");

    int array1[] = { 1, 2, 3, 4, 5, 6, 7 };
    int array2[] = { 6, 7, 8, 9, 10, 11 };
    int array3[] = { 11, 12, 13, 14, 15 };
    int size1 = 7;
    int size2 = 6;
    int size3 = 5;

    shiftArrayRight(array1, size1);
    shiftArrayRight(array2, size2);
    shiftArrayRight(array3, size3);

    printArray("Массив 1 после сдвига вправо: ", array1, size1);
    printArray("Массив 2 после сдвига вправо: ", array2, size2);
    printArray("Массив 3 после сдвига вправо: ", array3, size3);
}
```



```

#include <stdio.h>
#include <locale.h>

void binaryAddition(const int* array1, const int* array2, int*
result, int size) {
    int carry = 0;
    for (int i = size - 1; i >= 0; i--) {
        int sum = array1[i] + array2[i] + carry;
        result[i] = sum % 2;
        carry = sum / 2;
    }
}

void printArray(const char* message, int* array, int size) {
    printf("%s", message);
    for (int i = 0; i < size; i++) {
        printf("%d ", array[i]);
    }
    printf("\n");
}

int main() {
    setlocale(LC_ALL, "RU_ru");

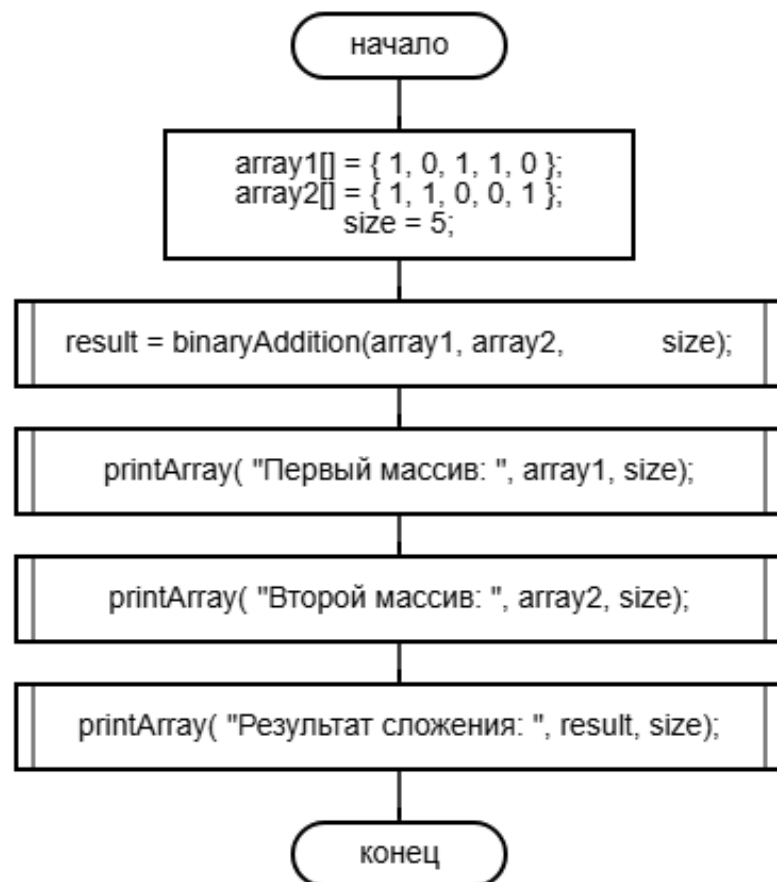
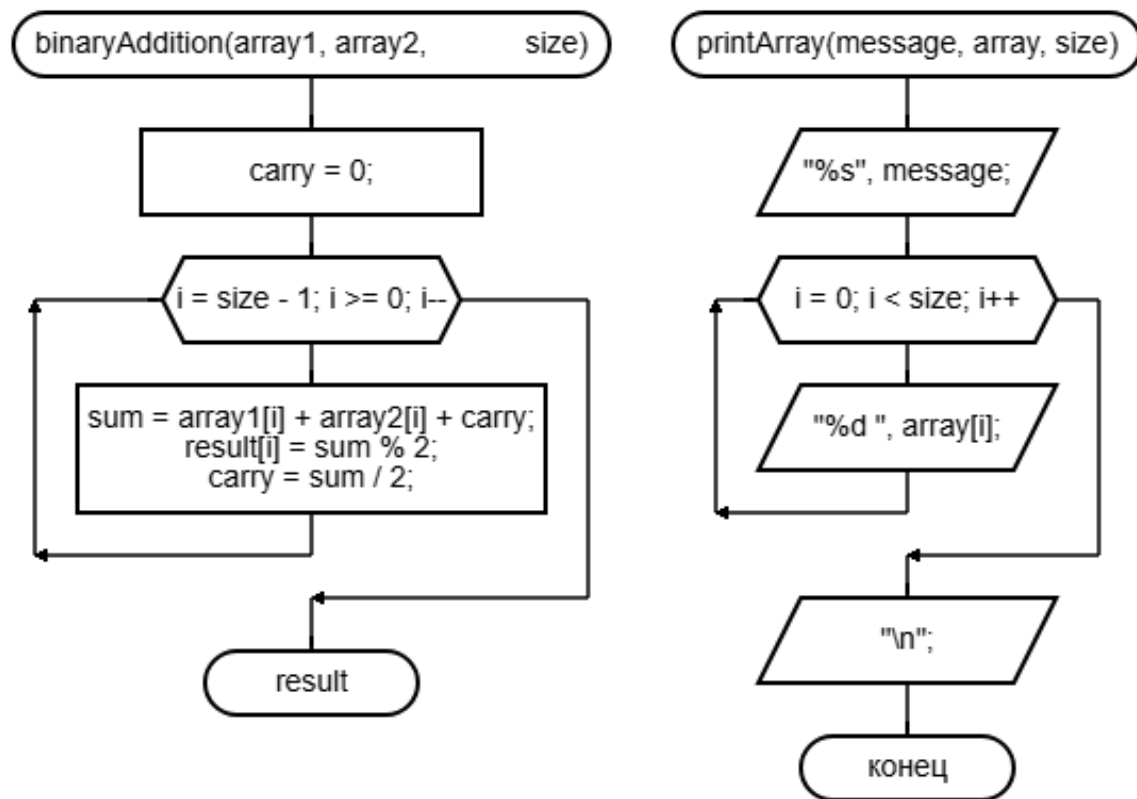
    int array1[] = { 1, 0, 1, 1, 0 };
    int array2[] = { 1, 1, 0, 0, 1 };
    int result[5];
    int size = 5;

    binaryAddition(array1, array2, result, size);

    printArray("Первый массив: ", array1, size);
    printArray("Второй массив: ", array2, size);
    printArray("Результат сложения: ", result, size);

    return 0;
}

```



```

#include <stdio.h>
#include <locale.h>

void binaryMultiplication(const int* num1, const int* num2, int*
result, int size) {
    for (int i = size - 1; i >= 0; i--) {
        if (num2[i] == 1) {
            int carry = 0;
            for (int j = size - 1; j >= 0; j--) {
                int product = num1[j] * num2[i] + result[i + j + 1]
+ carry;
                result[i + j + 1] = product % 2;
                carry = product / 2;
            }
            result[i] += carry;
        }
    }
}

void printArray(const char* message, int* array, int size) {
    printf("%s", message);
    for (int i = 0; i < size; i++) {
        printf("%d ", array[i]);
    }
    printf("\n");
}

int main() {
    setlocale(LC_ALL, "RU_ru");

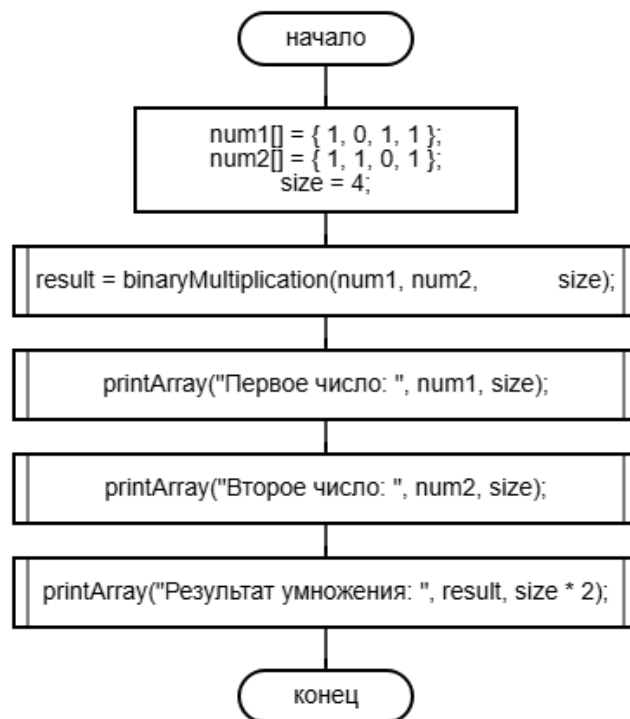
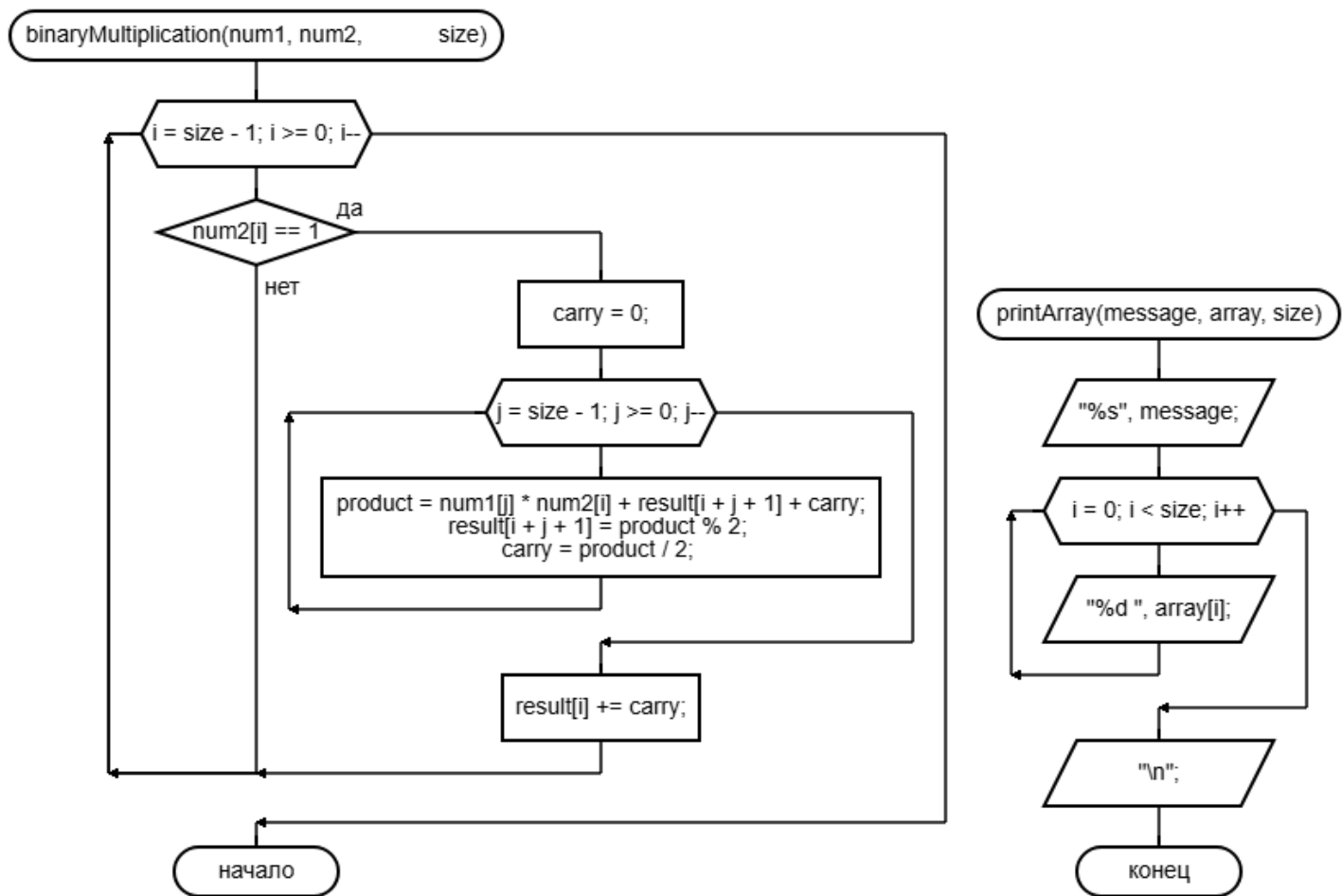
    int num1[] = { 1, 0, 1, 1 };
    int num2[] = { 1, 1, 0, 1 };
    int size = 4;
    int result[8] = { 0 };

    binaryMultiplication(num1, num2, result, size);

    printArray("Первое число: ", num1, size);
    printArray("Второе число: ", num2, size);
    printArray("Результат умножения: ", result, size * 2);

    return 0;
}

```



```

#include <stdio.h>
#include <locale.h>
#include <math.h>

int binaryFloatAddition(int* mantissa1, int* mantissa2, int exp1,
int exp2, int* resultMantissa, int size) {
    int expDiff = exp1 - exp2;
    int resultExp;

    if (expDiff > 0) {
        for (int i = size - 1; i >= expDiff; i--) {
            mantissa2[i] = mantissa2[i - expDiff];
        }
        for (int i = 0; i < expDiff; i++) {
            mantissa2[i] = 0;
        }
        resultExp = exp1;
    }
    else {
        for (int i = size - 1; i >= -expDiff; i--) {
            mantissa1[i] = mantissa1[i + expDiff];
        }
        for (int i = 0; i < -expDiff; i++) {
            mantissa1[i] = 0;
        }
        resultExp = exp2;
    }

    int carry = 0;
    for (int i = size - 1; i >= 0; i--) {
        int sum = mantissa1[i] + mantissa2[i] + carry;
        resultMantissa[i] = sum % 2;
        carry = sum / 2;
    }

    if (carry == 1) {
        for (int i = size - 1; i > 0; i--) {
            resultMantissa[i] = resultMantissa[i - 1];
        }
        resultMantissa[0] = 1;
        resultExp++;
    }

    return resultExp;
}

```

```

void printArray(const char* message, int* array, int size) {
    printf("%s", message);
    for (int i = 0; i < size; i++) {
        printf("%d", array[i]);
    }
    printf("\n");
}

int main() {
    setlocale(LC_ALL, "RU_ru");

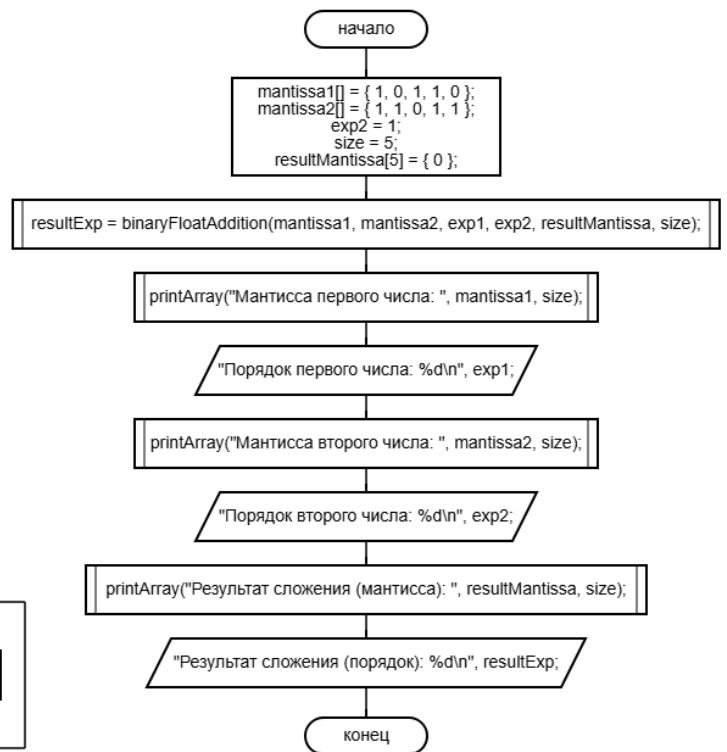
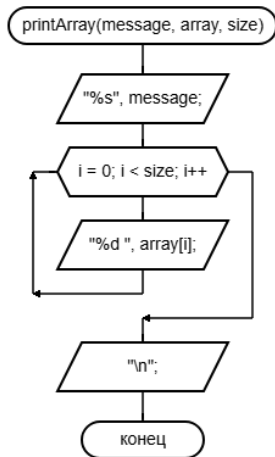
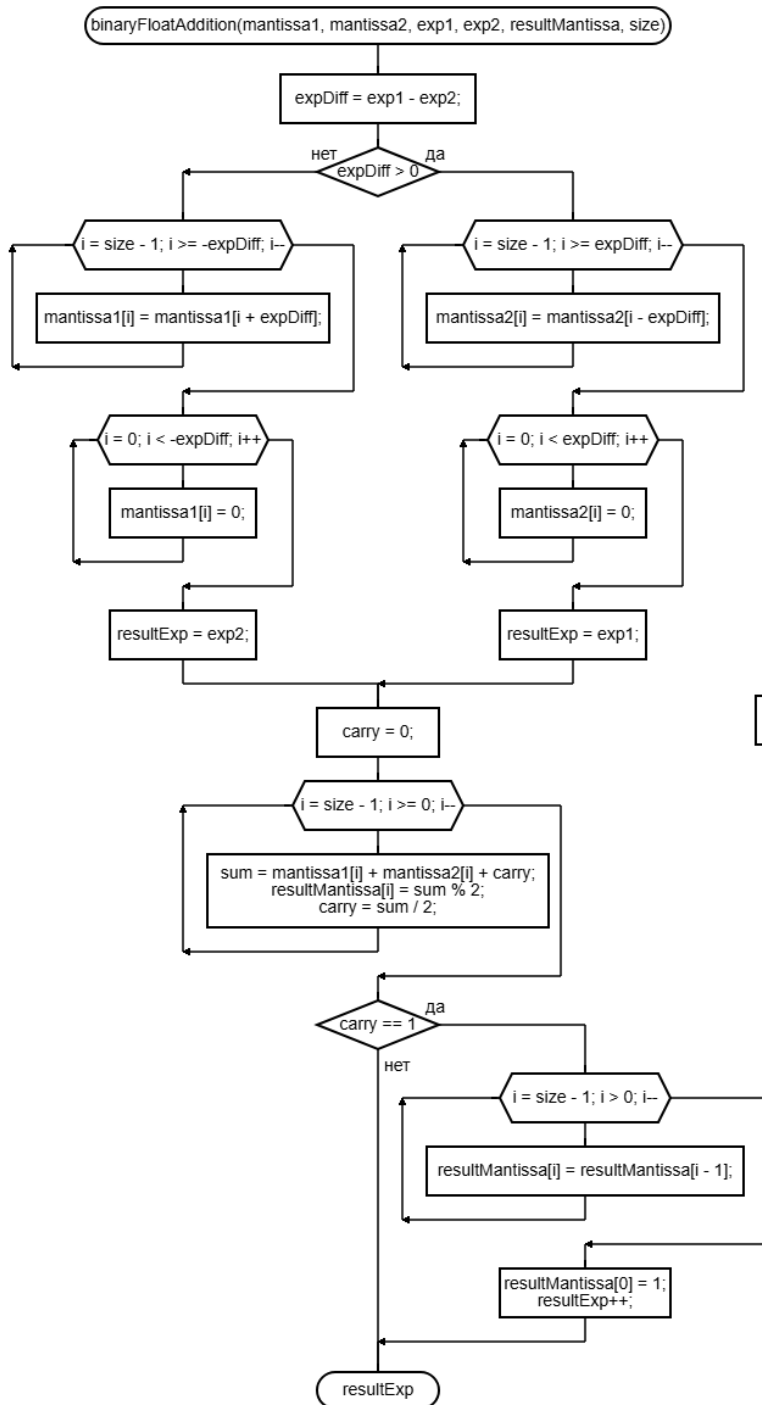
    int mantissa1[] = { 1, 0, 1, 1, 0 };
    int mantissa2[] = { 1, 1, 0, 1, 1 };
    int exp1 = 3;
    int exp2 = 1;
    int size = sizeof(mantissa1) / sizeof(mantissa1[0]);
    int resultMantissa[5] = { 0 };

    int resultExp = binaryFloatAddition(mantissa1, mantissa2, exp1,
exp2, resultMantissa, size);

    printArray("Мантисса первого числа: ", mantissa1, size);
    printf("Порядок первого числа: %d\n", exp1);
    printArray("Мантисса второго числа: ", mantissa2, size);
    printf("Порядок второго числа: %d\n", exp2);
    printArray("Результат сложения (мантисса): ", resultMantissa,
size);
    printf("Результат сложения (порядок): %d\n", resultExp);

    return 0;
}

```


```

#include <stdio.h>
#include <locale.h>

void invertBits(int* array, int* invertedArray, int size) {
    for (int i = 0; i < size; i++) {
        invertedArray[i] = array[i] ^ 1;
    }
}

void addOne(int* array, int size) {
    int carry = 1;
    int i = size - 1;
    while (i >= 0 && carry > 0) {
        int sum = array[i] + carry;
        array[i] = sum % 2;
        carry = sum / 2;
        i--;
    }
}

void printArray(const char* message, int* array, int size) {
    printf("%s", message);
    for (int i = 0; i < size; i++) {
        printf("%d", array[i]);
    }
    printf("\n");
}

int main() {
    setlocale(LC_ALL, "RU_ru");

    int number[] = { 0, 1, 1, 0, 1 };
    int size = 5;
    int invertedNumber[5] = { 0 };

    invertBits(number, invertedNumber, size);
    addOne(invertedNumber, size);

    printArray("Исходное число: ", number, size);
    printArray("Число в дополнительном коде: ", invertedNumber,
size);

    return 0;
}

```

