

Odwzorowanie bitowe

sprawozdanie

przedmiot: Metody i języki programowania

1. Cel ćwiczenia

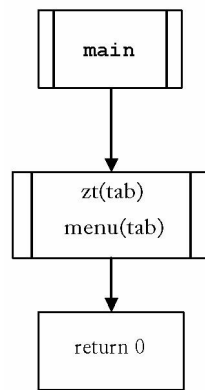
Celem ćwiczenia było napisanie programu umożliwiającego wprowadzanie i odczytywanie do/z tablicy jednowymiarowej podanego zbioru liczb całkowitych.

Zapis i odczyt danych zorganizowany jest w postaci tzw. odwzorowania bitowego.

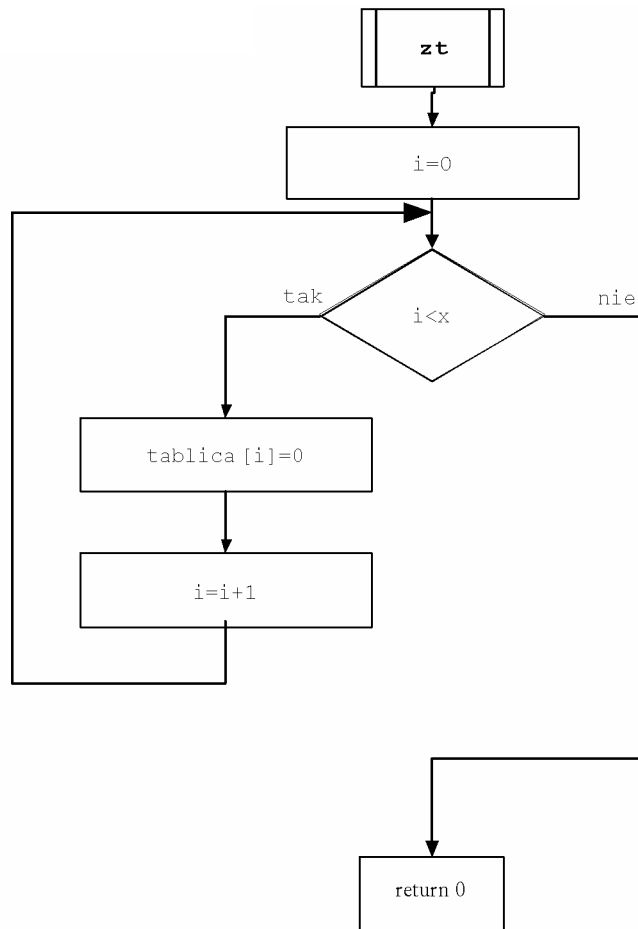
Zamiast przechowania 10.000 kolejnych liczb całkowitych na typie *integer* (w zależności od środowiska, 16 bitowego, na 2 bajtach, czy 32 bitowego, czyli na 4 bajtach), gdzie potrzebować będziemy 40.000 bajtów, można to zapisać bitowo jako ciąg 10.000 bitów, gdzie i -ty bit jedynką wtedy i tylko wtedy, gdy liczba i znajduje się w zbiorze.

2. Schematy blokowe

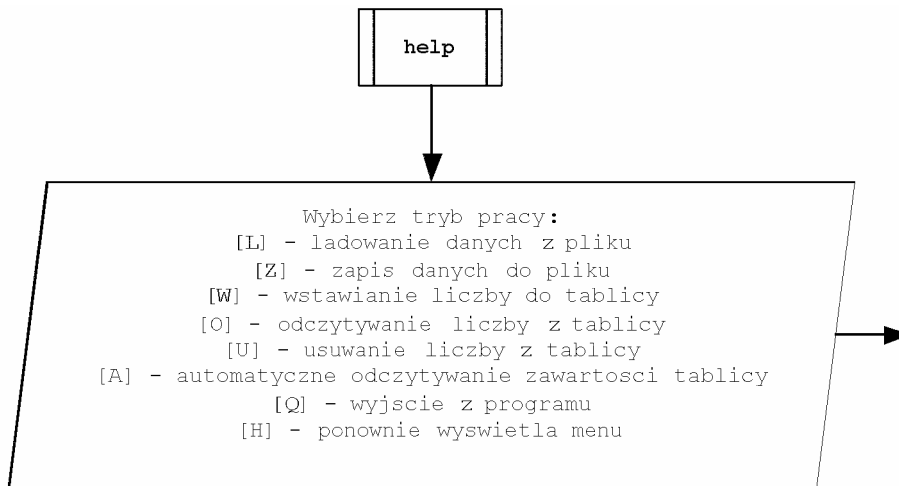
Funkcja główna **main**



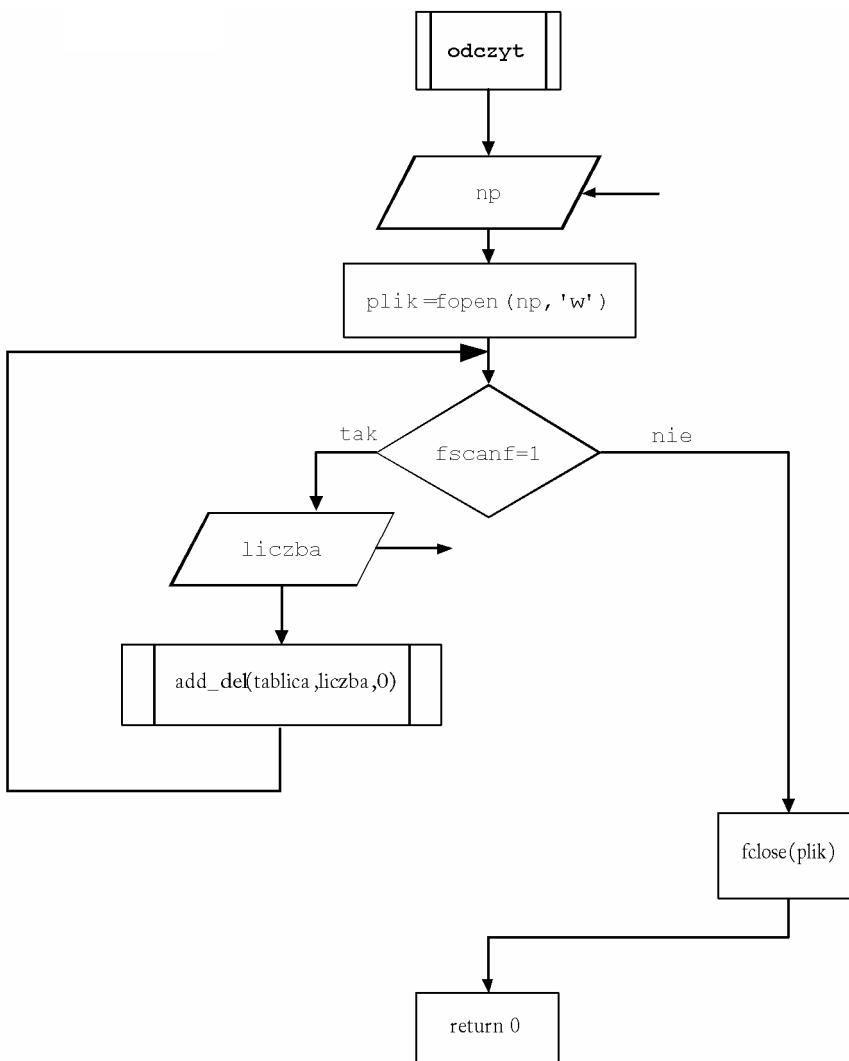
Funkcja **zt** (zerowanie tablicy)



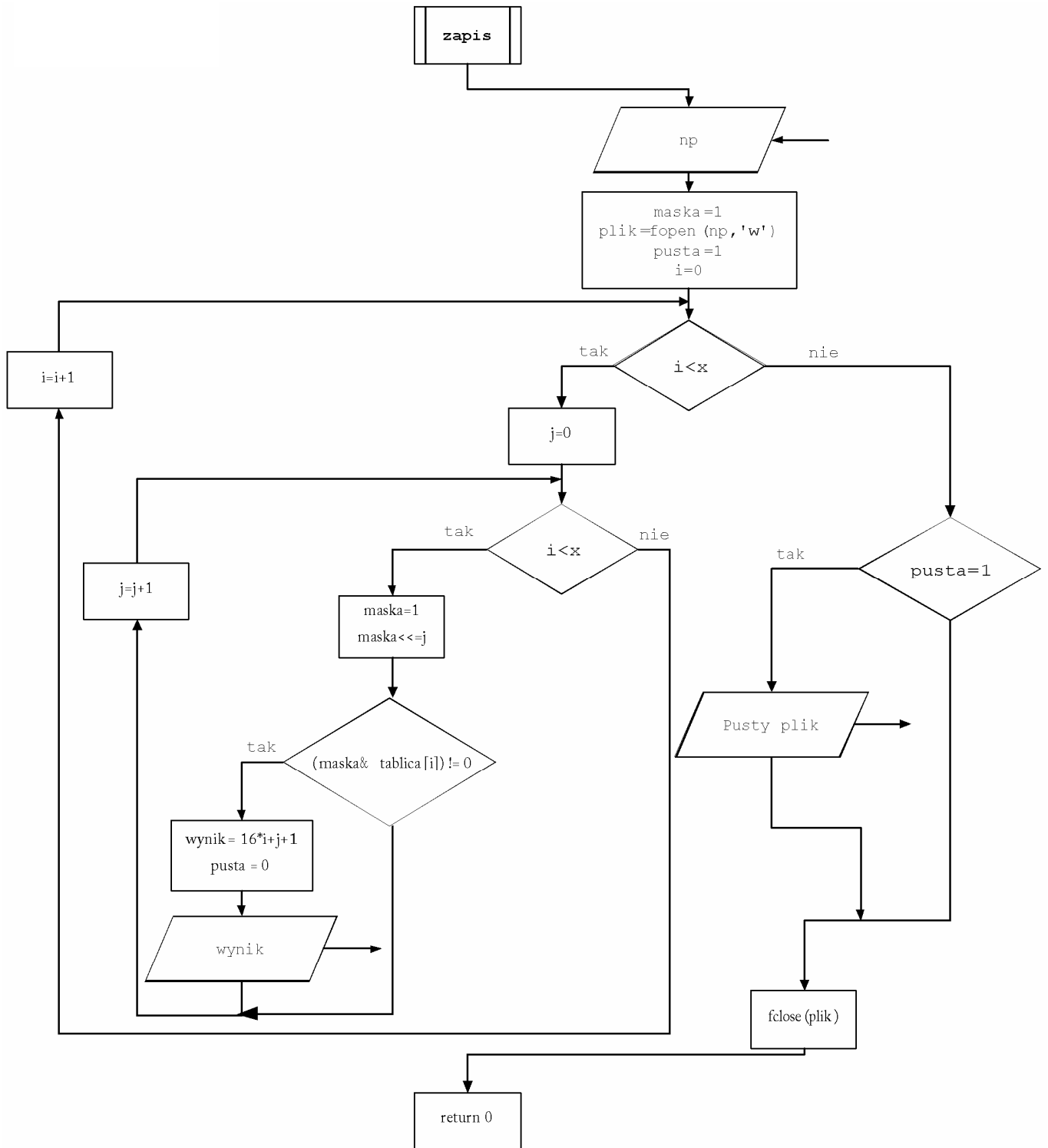
Funkcja help (wyświetla menu z listą opcji do wyboru)



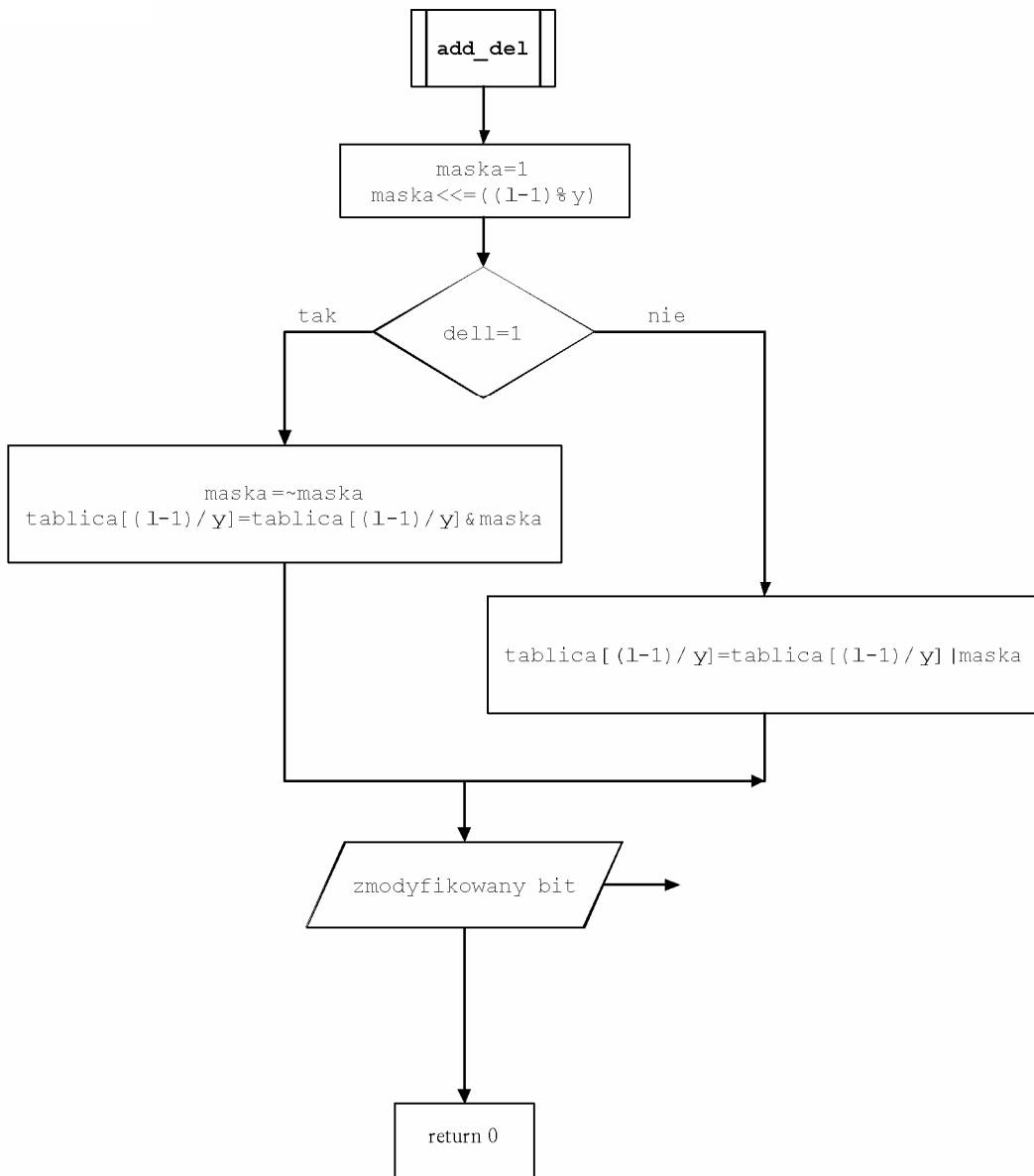
Funkcja odczyt (odczyt liczb z pliku i zapisanie ich do tablicy)



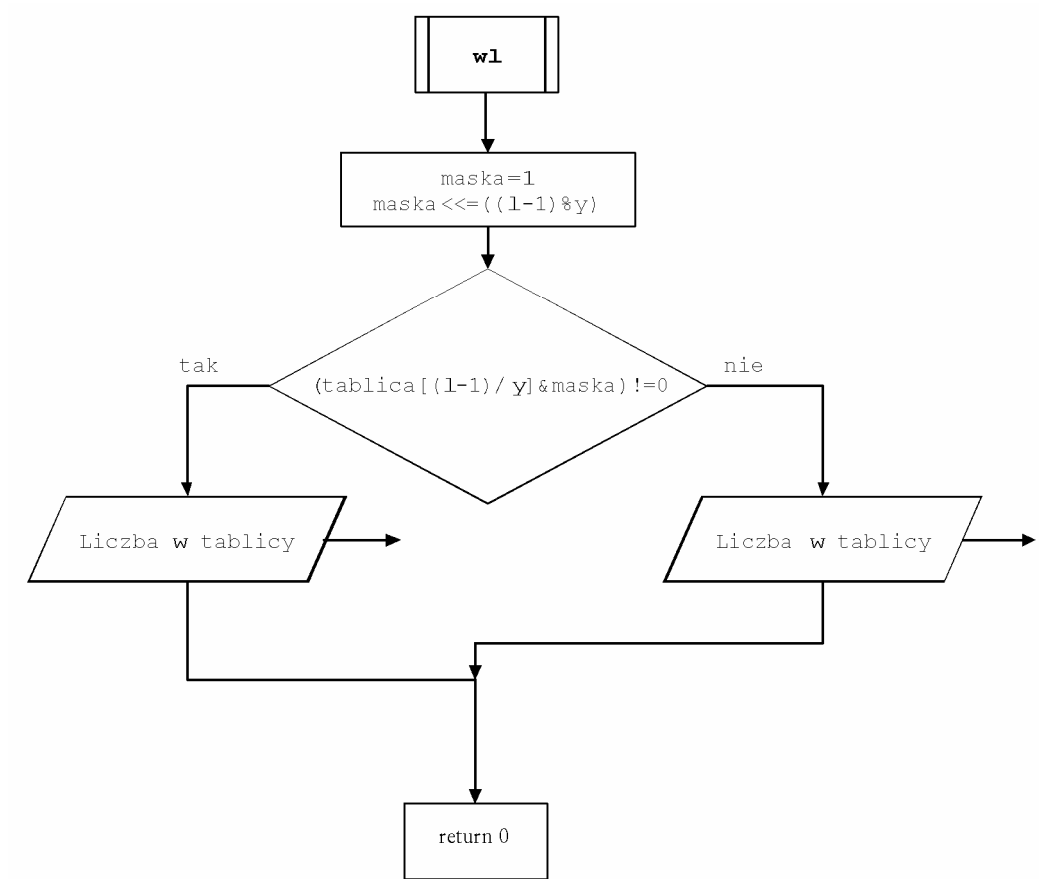
Funkcja zapis (zapis liczb z tablicy do pliku)



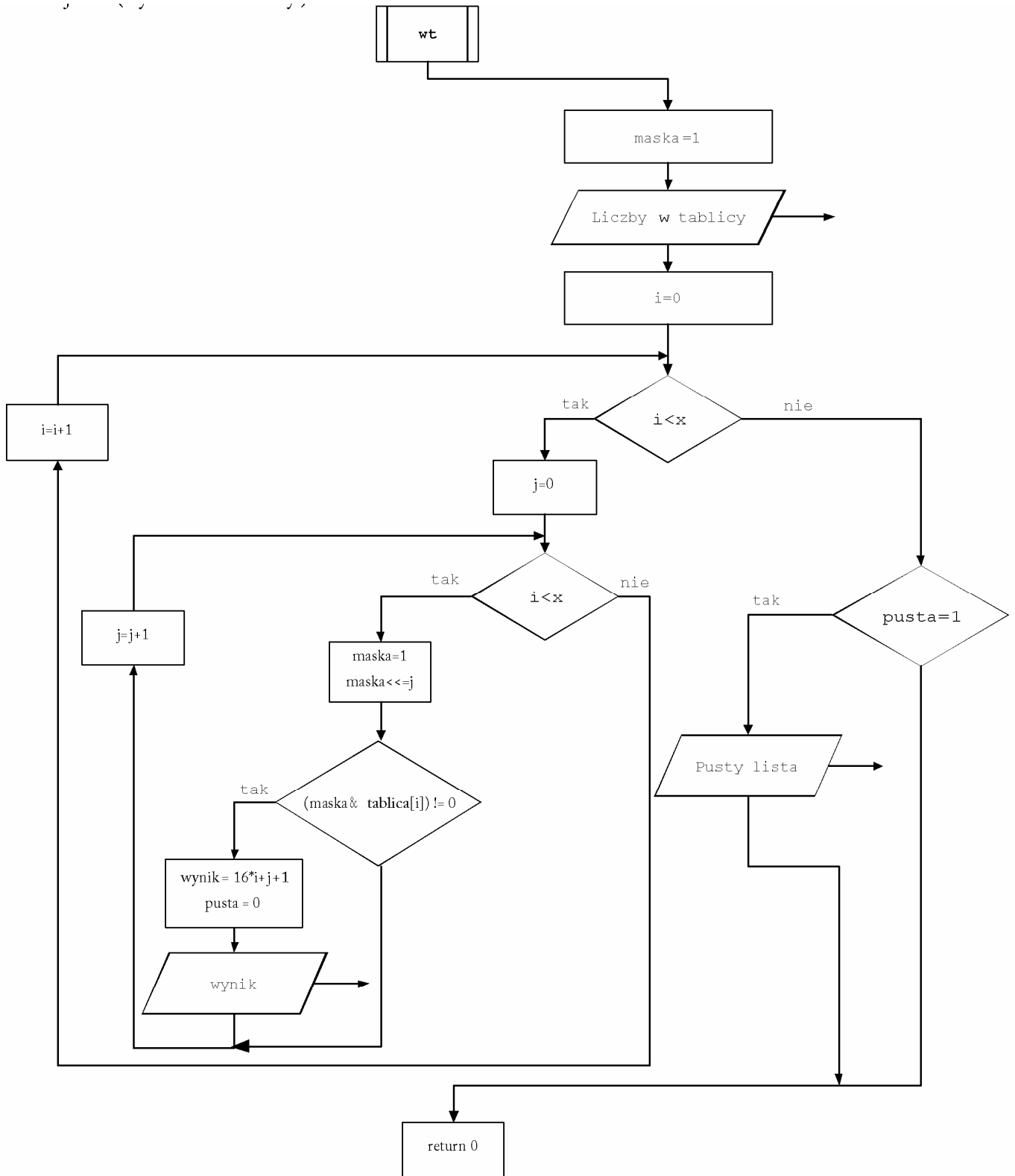
Funcja `add_del` (dodanie/usunięcie liczby do tablicy)



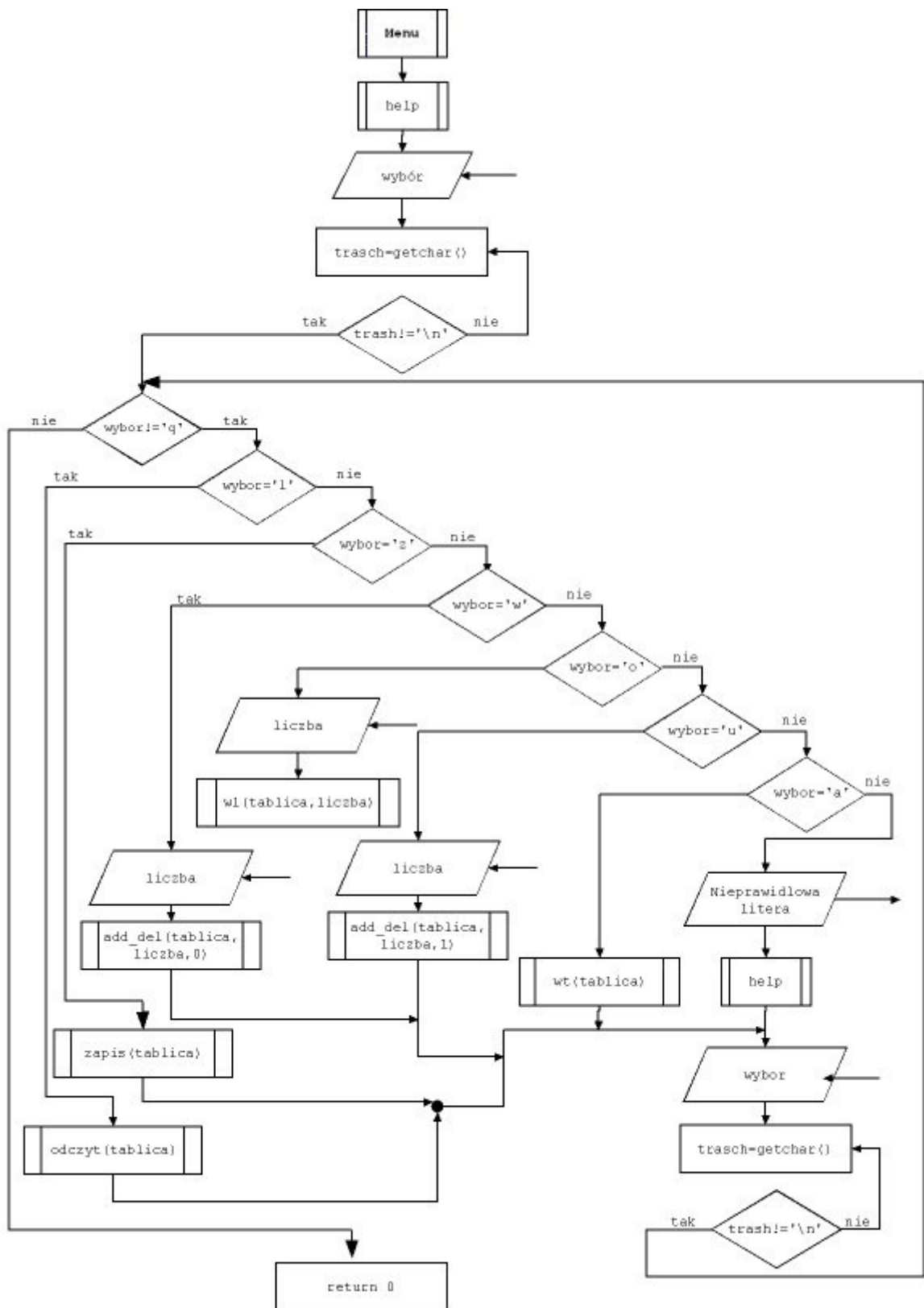
Funkcja w1 (sprawdza i wyświetla z tablicy podaną liczbę)



Funkcja wt (wyświetla wszystkie liczby znajdujące się w tablicy)



Fukcja menu (odpowiada za „sterowanie” programem)



3. Kod źródłowy programu

```
#include <stdio.h>
#include <stdlib.h>
#define y (sizeof(int)*8)
#define x (10000 / y) + 1

main ()
{
    unsigned int tab[x];
    zt(tab);
    menu(tab);
    return 0;
}

//zt - zerowanie tablicy
int zt(unsigned int *tablica)
{
    int i;
    for(i =0; i < x; i++)
        tablica[i] = 0;
    return 0;
}

//wt - wyswietlanie tablicy
int wt(unsigned int *tablica)
{
    int i, j, wynik, pusta=1;
    unsigned int maska = 1;
    printf("\nWyswietlone zostaly liczby znajdujace sie w tablicy:\n");
    for(i = 0; i < x; i++)
    {
        for(j = 0; j < y; j++)
        {
            maska = 1;
            maska <<= j;
            if((maska & tablica[i]) != 0)
            {
                wynik = y*i+j+1;
                printf("%u\n",wynik);
                pusta=0;
            }
        }
    }
    if(pusta == 1) printf("Lista jest pusta.\n\n");
    return 0;
}

//wl - wyswietlanie liczby
int wl(unsigned int *tablica, int l)
{
    unsigned int maska = 1;
    maska <<= ((l - 1) % y);
    if((tablica[(l - 1) / y] & maska) != 0)
        printf("\nLiczba znajduje sie w tablicy.\n");
    else
        printf("Nie ma takiej liczby w tablicy.\n\n");
    return 0;
}
```

```
// dodanie_usuniecie liczby

int add_del(unsigned int *tablica,int l, int del)
{
    unsigned int maska;
    maska=1;
    maska <<= ((l - 1) % y);
    if (del == 1)
    {
        maska = ~maska;
        tablica[(l - 1) / y] = tablica[(l - 1) / y] & maska;
    } else
        tablica[(l - 1) / y] = tablica[(l - 1) / y] | maska;
    printf("Bit ");
    printf("%u",((l - 1) % y));
    printf(" w komorce tab[");
    printf("%u",((l - 1) / y));
    printf("] zostal zmodyfikowany\n");
    return 0;
}

// odczyt- odczyt z pliku

odczyt(unsigned int *tablica)
{
    int liczba;
    char np[20];
    FILE *plik;
    printf("Podaj nazwe pliku: ");
    scanf("%s",np);
    printf("\n");
    plik = fopen(np, "r");

    while ( fscanf(plik,"%u ",&liczba) == 1)
    {
        printf("Do tablicy wpisano liczbe: %u \n", liczba);
        add_del(tablica, liczba, 0);
        printf("\n");
    }

    fclose(plik);
}

// zapis - zapis do pliku
zapis(unsigned int *tablica)
{
    int i, j, liczba, wynik, pusta;
    char np[20];
    unsigned int maska = 1;
    FILE *plik;
    printf("Podaj nazwe pliku: ");
    scanf("%s",np);
    printf("\n");
    plik = fopen(np, "w");
    printf("Do pliku %s wpisane nastepujace liczby: \n",np);
    pusta = 1;
    for(i = 0; i < x; i++)
    {
        for(j = 0; j < y; j++)
        {
            maska = 1;
            maska <<= j;
            if((maska & tablica[i]) != 0)
            {
                wynik = 16*i+j+1;
                printf("%u ",wynik);
                fprintf(plik,"%u ",wynik);
                pusta = 0;
            }
        }
    }
    if(pusta == 1) printf("W tablicy nie bylo liczb. Plik jest pusty.\n");
    fclose(plik);
    return 0;
}
```

```
void help(void)
{
    printf("Wybierz tryb pracy:\n");
    printf("[L] - ladowanie danych z pliku\n");
    printf("[Z] - zapis danych do pliku\n");
    printf("[W] - wstawianie liczby do tablicy\n");
    printf("[O] - odczytywanie liczby z tablicy\n");
    printf("[U] - usuwanie liczby z tablicy\n");
    printf("[A] - automatyczne odczytywanie zawartosci tablicy\n");
    printf("[Q] - wyjscie z programu\n");
    printf("[H] - ponownie wyswietla menu\n\n");
}
```

```
int menu(unsigned int *tablica)
{
    char wybor, trash;
    int liczba;
    help();
    scanf("%c", &wybor);
    do
    {
        trash = getchar();
    } while (trash != '\n');

    while(wybor != 'q')
    {
        switch (wybor)
        {
            case 'l':
            {
                odczyt(tablica);
                break;
            }
            case 'z':
            {
                zapis(tablica);
                break;
            }
            case 'w':
            {
                printf("Podaj liczbe do wstawienia: \n");
                scanf("%u", &liczba);
                add_del(tablica, liczba, 0);
                break;
            }
            case 'o':
            {
                printf("Podaj liczbe do sprawdzenia:\n");
                scanf("%u", &liczba);
                wl(tablica, liczba);
                break;
            }
            case 'u':
            {
                printf("Podaj liczbe do usuniecia: ");
                scanf("%u", &liczba);
                add_del(tablica, liczba, 1);
                break;
            }
            case 'a':
            {
                wt(tablica);
                break;
            }
            default:
            {
                printf("\nNacisnales nieprawidlowa litere.\n");
                help();
            }
        }
    }
    printf("\nWybierz opcje:\n");
    scanf("%c", &wybor);
    do
    {
        trash = getchar();
    } while (trash != '\n');
}
return 0;
}
```

4. Wnioski

Program został napisany w sposób pozwalający na uruchamianie go niezależnie od użytej platformy sprzętowej. Znaczy to, że nie ma wpływu, czy typ *integer* będzie 2 bajtowy, czy jakikolwiek inny. Stało się to możliwe dzięki zadeklarowaniu stałej *y*, która przyjmuje wartość $\text{sizeof}(\text{int}) * 8$. Pod tym kątem program został sprawdzony w kompilatorze Borlanda pod system DOS, gdzie typ *integer* jest 2 bajtowy, jaki i w systemie Unix, gdzie typ *integer* jest 4 bajtowy.

Ponadto składnia kodu źródłowego pozwala uruchomić program w systemie Dos/Windows jaki i w Unixie.

Program może służyć do sortowania liczb z tablicy/pliku.