

### Zad.1

W zadaniu należało zaalokować pamięć dla 1 000 000, następnie zmierzyć czas potrzebny do ich wypełnienia danymi oraz czas potrzebny na wyszukanie rekordu o indeksie 999 999.

Struktura rekordu:

```
struct Rec {  
    int id;          /* unikalny identyfikator, klucz główny */  
    char name[20];   /* nazwa */  
    char desc[90];   /* opis */  
};
```

Eksperyment przeprowadzono pięciokrotnie i wyciągnięto średnie wartości.  
Otrzymane dane zawiera tabela.

Czas wypełniania danych	0,099 s
Czas wyszukiwania rekordu	0,015 s
Zużycie pamięci	116 484 kB

### Zad.2

W zadaniu należało powtórzyć obliczenia z pierwszego zadania z wykorzystaniem bazy danych sqlite.

Obliczenia również wykonano pięciokrotnie i wyciągnięto średnią.

Struktura rekordu:

```
struct Rec {  
    int id;          /* unikalny identyfikator, klucz główny */  
    char name[8];     /* nazwa */  
    char desc[15];    /* opis */  
};
```

Czas wypełniania danych w pamięci	2,609 s
Czas wyszukiwania rekordu w pamięci	0,001 s
Zużycie pamięci	41 340 kB
Czas wypełniania danych w pliku	2,844 s
Czas wyszukiwania rekordu w pliku	0,001 s
Zużycie pamięci	6 412 kB

Struktura rekordu:

```
struct Rec {  
    int id;          /* unikalny identyfikator, klucz główny */  
    char name[18];   /* nazwa */  
    char desc[80];   /* opis */  
};
```

};

Czas wypełniania danych w pamięci	2,937 s
Czas wyszukiwania rekordu w pamięci	0,001 s
Zużycie pamięci	120 204 kB
Czas wypełniania danych w pliku	3,298 s
Czas wyszukiwania rekordu w pliku	0,001 s
Zużycie pamięci	6 412 kB

#### Wnioski:

Opierając się na zadaniu pierwszym można wyciągnąć wniosek iż operacje zapisu na pamięci programu są wykonywane znacznie szybciej niż w bazie danych bądź w pliku. Jednak jest to mało efektywny sposób ze względu na ograniczone zasoby.

Znacznie efektywniejsze wydaje się być użycie bazy danych sqlite. Z przebiegu laboratorium wynika, że baza sqlite zmniejsza zużycie pamięci potrzebnej do pracy z danymi, ponadto przyspiesza znacznie proces wyszukiwania rekordów.

Widać także, że zapisanie danych do pliku znacznie zmniejsza ilość pamięci potrzebnej do tego procesu, jednak wiąże się to z nieznacznie dłuższym czasem zapisu danych.

W programie przeprowadzającym obliczenia wykorzystano transakcje oraz predefiniowane zapytania. Dzięki temu znacznie przyspieszono działanie programu – około 100-krotnie.