Imię i nazwisko Marek Kubicki	Kierunek Informatyka techniczna ITE	Rok studiów i grupa 1 rok grupa 5
Data zajęć 19.01.2023	Numer i temat sprawozdania Lab 13. – Rekurencja, pliki i inne	

Cel:

• Opanowanie programowania z użyciem funkcji rekurencyjnych, wskaźników do funkcji oraz plików

Przebieg ćwiczeń:
Utworzyłem katalog roboczy lab_13 a następnie skopiowałem do niego odpowiednie pliki:

```
m@m-VirtualBox:~/Lab_13$ pwd
/home/m/Lab_13
m@m-VirtualBox:~/Lab_13$ ls
plik.asc  potega.c  skr1.asc
```

Następnie zacząłem wykonywać kolejne polecenia.
Pierwsze z nich polegało na stworzeniu funkcji power_int_rekur, która oblicza naturalne potęgi liczb całkowitych w sposób rekurencyjny. W tym celu wykorzystałem niżej przedstawiony warunek if()

```
assert(n>=0);
int potega=1;
if(n>0)
{
potega = potega * m * power_int_rekur(m, n-1);
}
```

Przykładowe potęgi obliczone przez tą funkcję:

```
rekurencyjnie:   2^0 =    1        -3^0 =      1
rekurencyjnie:   2^1 =    2        -3^1 =     -3
rekurencyjnie:   2^2 =    4        -3^2 =      9
rekurencyjnie:   2^3 =    8        -3^3 =    -27
```

Kolejna część zadania polegała na dodaniu opcji podawania danych przy wywoływaniu programu, W tym celu dodałem 2 argumenty wejścia do funkcji main :

```
int main (int argc, char** argv)
```

Z nich sczytywane są dane wejściowe a później obliczana jest odpowiednia potęga.

```
  int base = my_atoi( argv [1]);
  int exp = my_atoi( argv [2]);
  printf("Dane z linii poleceń: %d^%d=%d\n", base, exp, power_int_re-
kur(base,exp));
```

```
m@m-VirtualBox:~/Lab_13$ ./potega 5 3
Dane z linii poleceń: 5^3=125
```

Następne polecenie polegało na wprowadzeniu zmiennej statycznej do funkcji power_int_rekur która zliczałaby liczbę wywołeń tej funkcji (Wypis był wcześniej wykomentowany z powodu ilości wywołań >100):

```
  static int liczba_wywolan =0;
  liczba_wywolan++;
  printf("Wywołanie nr %d funkcji power_int_rekur\n", liczba_wywolan);
```

```
Wywołanie nr 4 funkcji power_int_rekur
Dane z linii poleceń: 5^3=125
Wywołanie nr 5 funkcji power_int_rekur
Wywołanie nr 6 funkcji power_int_rekur
rekurencyjnie:  2^0 =    1        -3^0 =      1
```

Ostatnie polecenie polegało na wprowadzeniu funkcjonalności odczytu danych z pliku. W tym celu stworzyłem odpowiedni plik, następnie dodałem do programu odpowiednie polecenia.
Poniższy kod pobiera adres pliku, sprawdza czy ów plik istnieje, a następnie sczytuje odpowiednie dane i oblicza potęgę.

```
  FILE *plik;
  plik = fopen("./plik", "r");
  if (plik == NULL)
  {
    printf("PLIK NIE ISTNIEJE LUB JEST PUSTY!!!");
    exit(0);
  }
  fscanf(plik, "%d %d", &base_f, &exp_f);
  fclose (plik);
  printf("Dane z pliku: %d^%d=%d\n", base_f, exp_f, power_int_re-
kur(base_f,exp_f));
```

```
Dane z pliku: 2^10=1024
```

Wnioski:

Rekurencja pozwala na uproszczenie struktury programu, a przez to na ułatwienie debugowania i rozszerzania istniejącego już kodu. Ponadto, rekurencja ułatwia zaimplementowanie niektórych algorytmów, poprzez rozdzielenie dużych problemów na wiele mniejszych.
Odczytywanie danych wejściowych z komendy uruchamiającej program jak i z plików pozwala na

stworzenie programów z ogromną ich ilością, przez to stworzone programy mogą wykonywać dużo bardzie skomplikowane zadania jak i na działanie z szybkością nie osiągalną w przypadku ręcznego wpisywania danych.