

<p align="center">Projekt podstawy programowania 2</p> <p align="center">Wydział Elektrotechniki Automatyki i Informatyki</p> <p align="center">Politechnika Świętokrzyska</p>	
Studia: Stacjonarne I stopnia	Kierunek: Informatyka
Data wykonania: 3.06.2016	Grupa: 1ID14A
Ocena:	1. Lisowski Wojciech 2. Łękawska Kinga
Temat projektu: <p align="center">Baza kontaktów</p>	

1. Temat projektu

Tematem projektu jest program „Baza kontaktów” stworzona z wykorzystaniem list. Baza posiada takie funkcje jak: dodawanie, usuwanie i edycja kontaktów złożonej z pól: ID, nazwisko, imię, adres (miasto, ulica, nr domu/mieszkania, kod pocztowy, poczta), telefon i e-mail. Kontakty można sortować po dowolnym polu rosnąco lub malejąco.

2. Realizacja projektu

2.1. Język programowania i środowisko:

Projekt został napisany w języku C w środowisku Code::Blocks na systemie Windows.

2.2. Załączone pliki nagłówkowe:

- stdio.h - obsługa standardowego wejścia/wyjścia.
- stdlib.h - standardowe narzędzia języka C
- string.h - obsługa ciągów znaków
- ctype.h - obsługa operacji na znakach

2.3. Struktura programu:

Projekt został podzielony na pliki źródłowe i nagłówkowe:

- main.c – plik źródłowy zawiera główne menu i wywoływane są w nim wszystkie funkcje z innych plików.
- add_and_remove_elements.c – plik źródłowy zawiera funkcję dodające i usuwające elementy z listy.
- operations_on_lists.c – plik źródłowy zawiera funkcję operacji na listach np. wyświetlanie.
- read_and_save_to_file.c – plik źródłowy zawiera funkcję operacji na plikach.
- sort_list.c – plik źródłowy zawiera funkcje sortujące listę.

- baza_kontaktow.h – plik nagłówkowy zawiera nagłówki funkcji ze wszystkich plików programu.

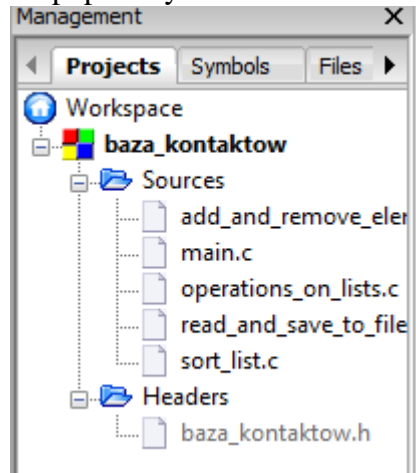
3. Kompilacja programu

Kompilację należy przeprowadzić w programie Code::Blocks.

3.1. Proces kompilacji:

- Uruchomienie pliku baza_kontaktow.cbp za pomocą programu Code::Blocks.

Po poprawnym otwarciu struktura programu powinna wyglądać tak:



- Następnie w programie Code::Blocks przechodzimy do zakładki „Build” i klikamy opcję „Build and run” lub klikamy klawisz F9.
- Po prawidłowej kompilacji powinno wyświetlić się główne menu programu:

```

C:\Users\Wojtek\Documents\C\PP2\projektp2\baza_kontaktow\bin\Debug\baza_kontaktow.exe
Baza kontaktow v1.28
##### MENU #####
1.Dodaj osobe
2.Usun osobe
3.Edytuj osobe
4.Wyświetl baze
5.Wyświetl kontakt
6.ZAPIS DO PLIKU
7.SORTOWANIE
8.WYJSCIE i USUNIECIE LISTY

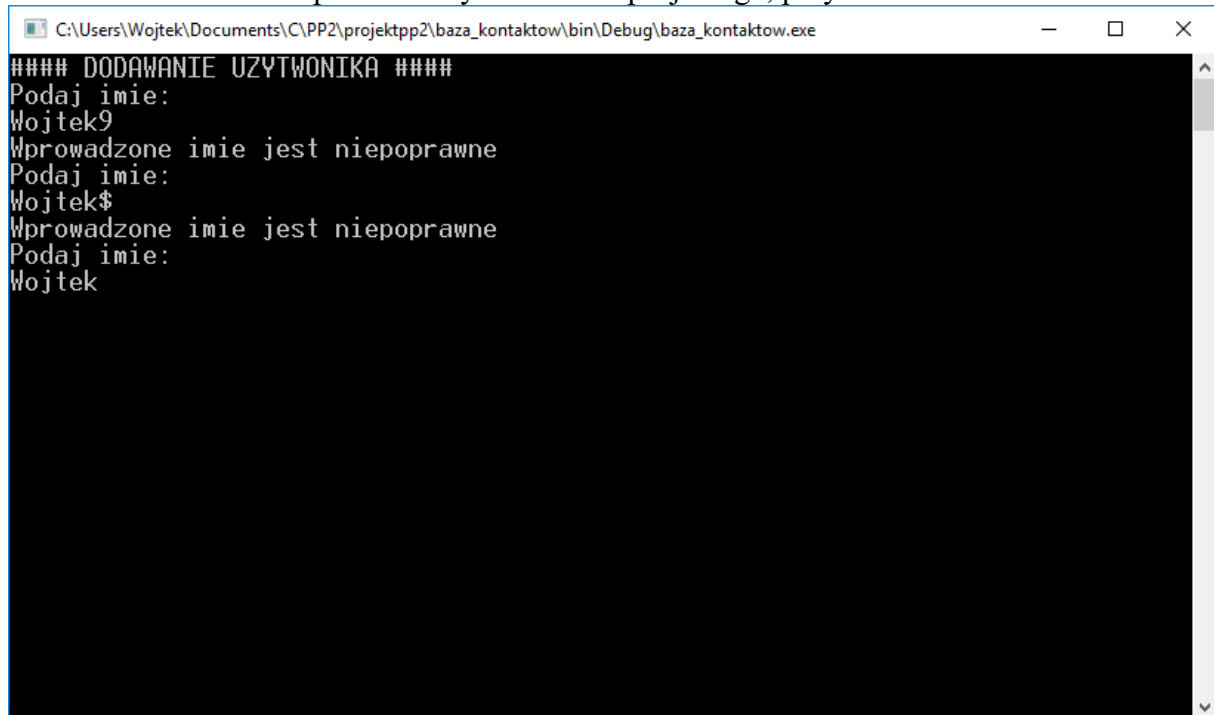
```

4. Przykładowe działanie programu

4.1. Dodawanie kontaktu

Aby dodać kontakt wybieramy opcję nr 1 w menu głównym.

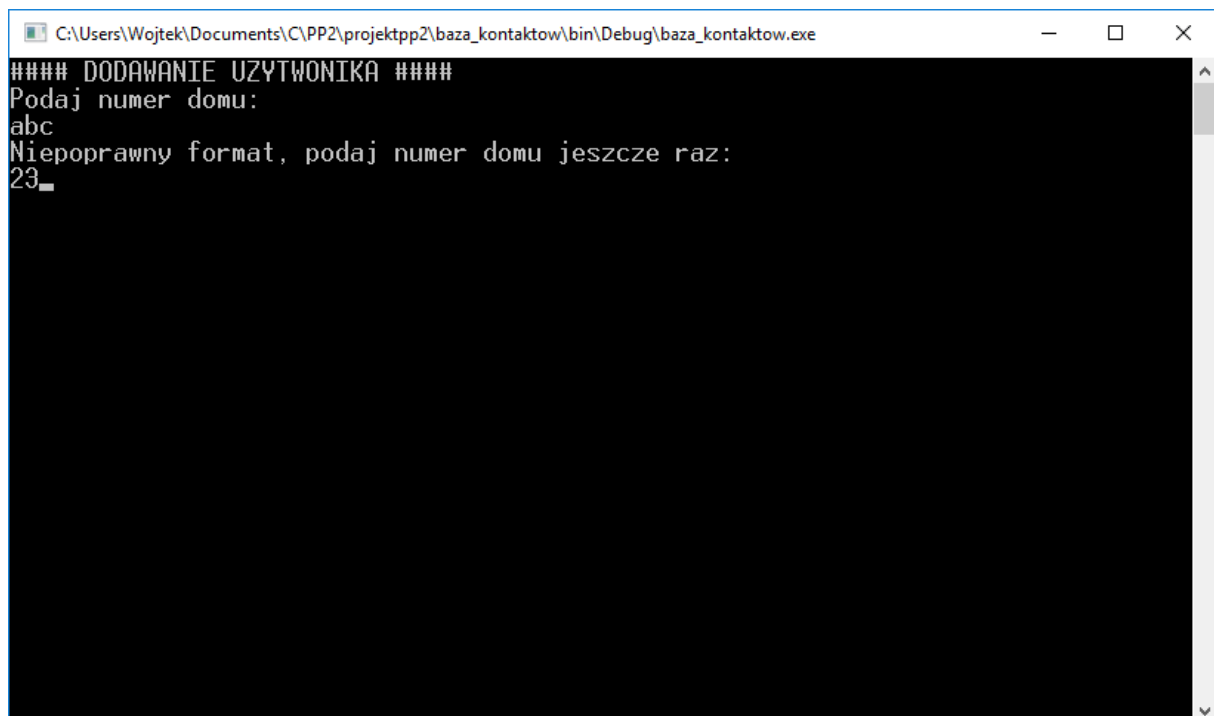
Wprowadzane dane są sprawdzane pod kątem poprawności np. w imieniu nie można podać liczby ani znaku specjalnego, przykład:



```
C:\Users\Wojtek\Documents\C\PP2\projektpp2\baza_kontaktow\bin\Debug\baza_kontaktow.exe
#### DODAWANIE UZYTOWNIKA ####
Podaj imie:
Wojtek9
Wprowadzone imie jest niepoprawne
Podaj imie:
Wojtek$
Wprowadzone imie jest niepoprawne
Podaj imie:
Wojtek
```

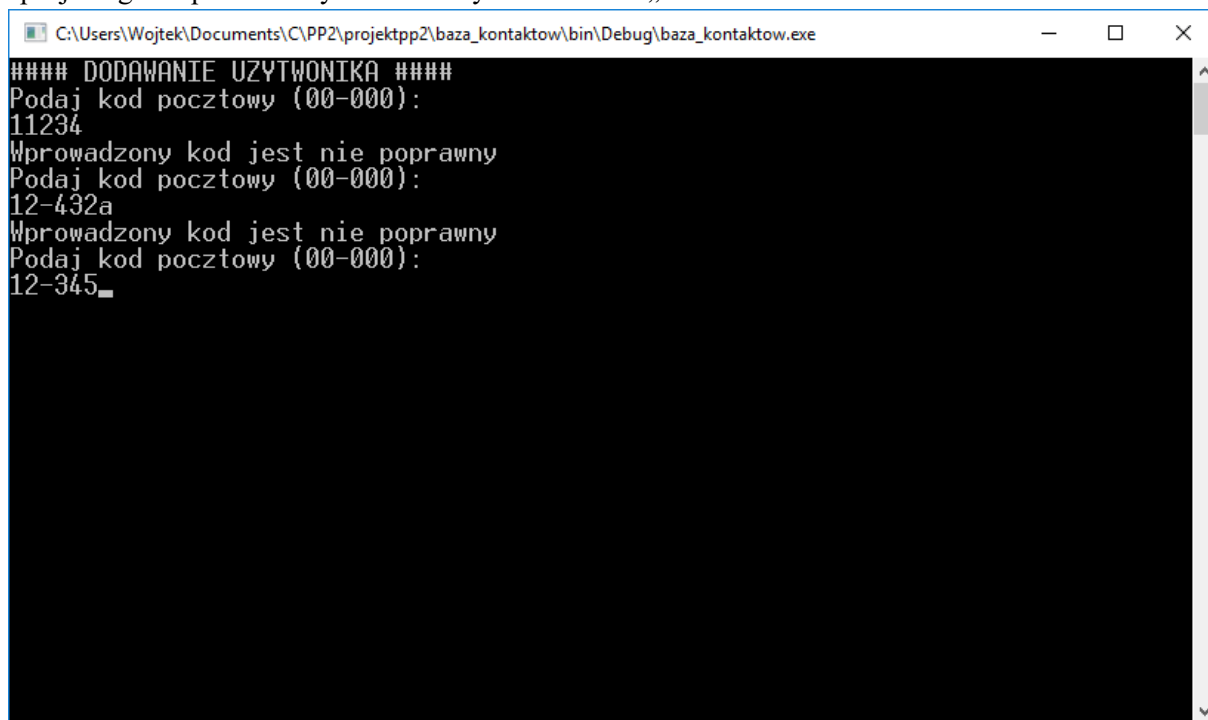
Taka walidacja działa tak samo dla pól: nazwisko, miasto, ulica i poczta.

Dla pola „numer domu” nie można wprowadzać znaków specjalnych ani liter:



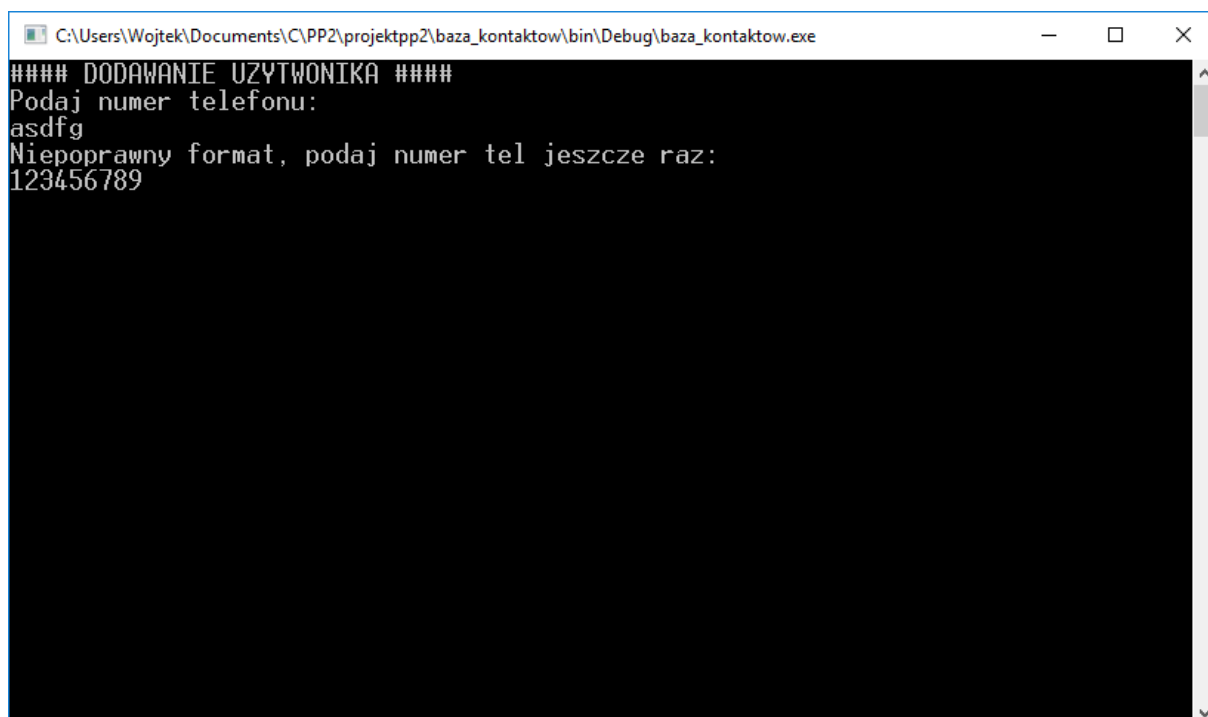
```
C:\Users\Wojtek\Documents\C\PP2\projektpp2\baza_kontaktow\bin\Debug\baza_kontaktow.exe
#### DODAWANIE UZYTOWNIKA ####
Podaj numer domu:
abc
Niepoprawny format, podaj numer domu jeszcze raz:
23_
```

Pole „kod pocztowy” także posiada walidację danych, nie można wprowadzić litery ani znaku specjalnego i wprowadzany kod musi być w formacie „00-000”:



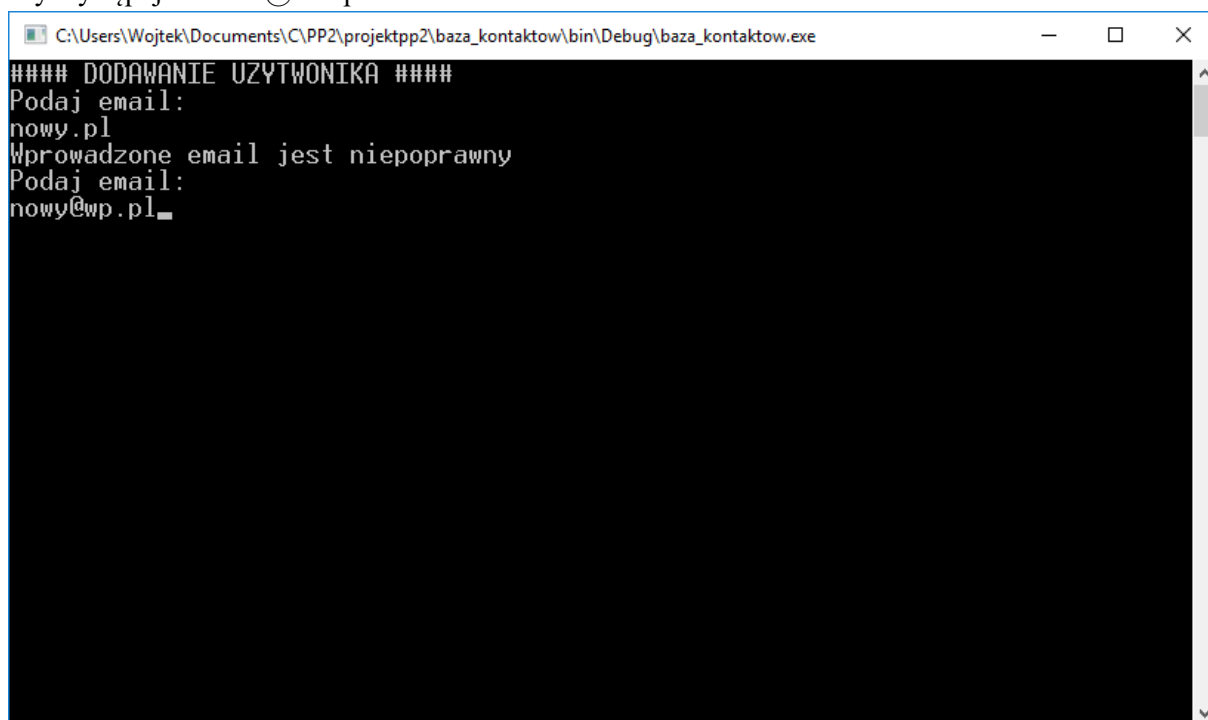
```
C:\Users\Wojtek\Documents\C\PP2\projektpp2\baza_kontaktow\bin\Debug\baza_kontaktow.exe
#### DODAWANIE UZYTOWNIKA ####
Podaj kod pocztowy (00-000):
11234
Wprowadzony kod jest nie poprawny
Podaj kod pocztowy (00-000):
12-432a
Wprowadzony kod jest nie poprawny
Podaj kod pocztowy (00-000):
12-345_
```

W polu „numer telefonu” można podać kilka numerów telefonów, numery są sprawdzane pod kątem poprawności np.:



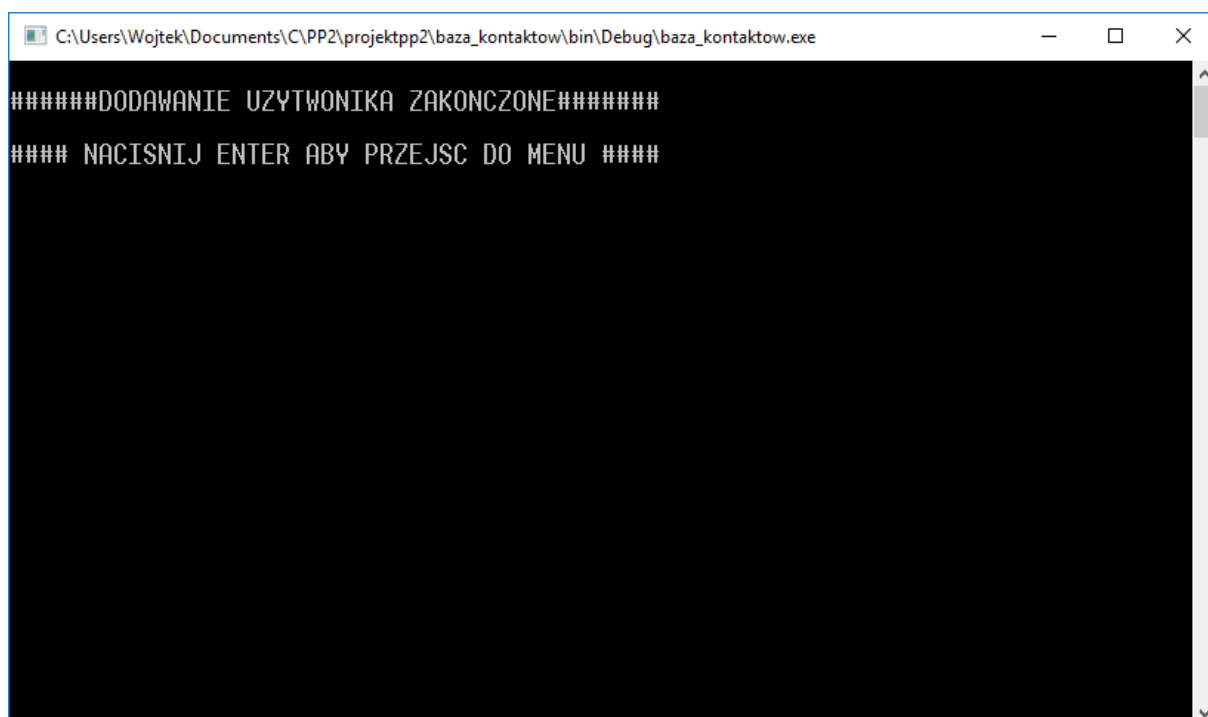
```
C:\Users\Wojtek\Documents\C\PP2\projektpp2\baza_kontaktow\bin\Debug\baza_kontaktow.exe
#### DODAWANIE UZYTOWNIKA ####
Podaj numer telefonu:
asdfg
Niepoprawny format, podaj numer tel jeszcze raz:
123456789
```

Ostatnim polem jest „adres e-mail”. Adresów e-mail można podać kilka. Każdy email jest sprawdzany czy występuje w nim @ i . np.:



```
C:\Users\Wojtek\Documents\C\PP2\projektp2\baza_kontaktow\bin\Debug\baza_kontaktow.exe
#### DODAWANIE UZYTOWNIKA ####
Podaj email:
nowy.pl
Wprowadzone email jest niepoprawny
Podaj email:
nowy@wp.pl
```

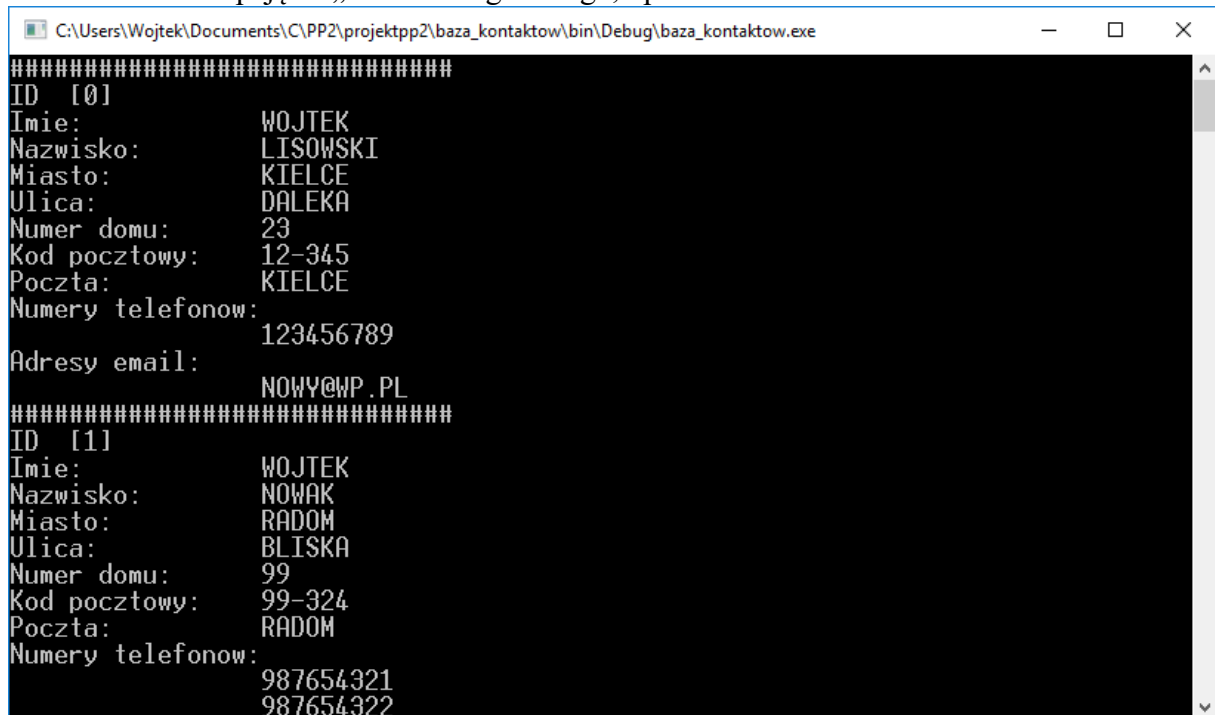
Na koniec dostajemy komunikat:



```
C:\Users\Wojtek\Documents\C\PP2\projektp2\baza_kontaktow\bin\Debug\baza_kontaktow.exe
#####DODAWANIE UZYTOWNIKA ZAKONCZONE#####
#### NACISNIJ ENTER ABY PRZEJSC DO MENU ####
```

4.2. Wyświetlanie bazy kontaktów.

Aby wyświetlić dodanych lub wczytanych z pliku użytkowników wybieramy opcję nr „4” z menu głównego, np.:



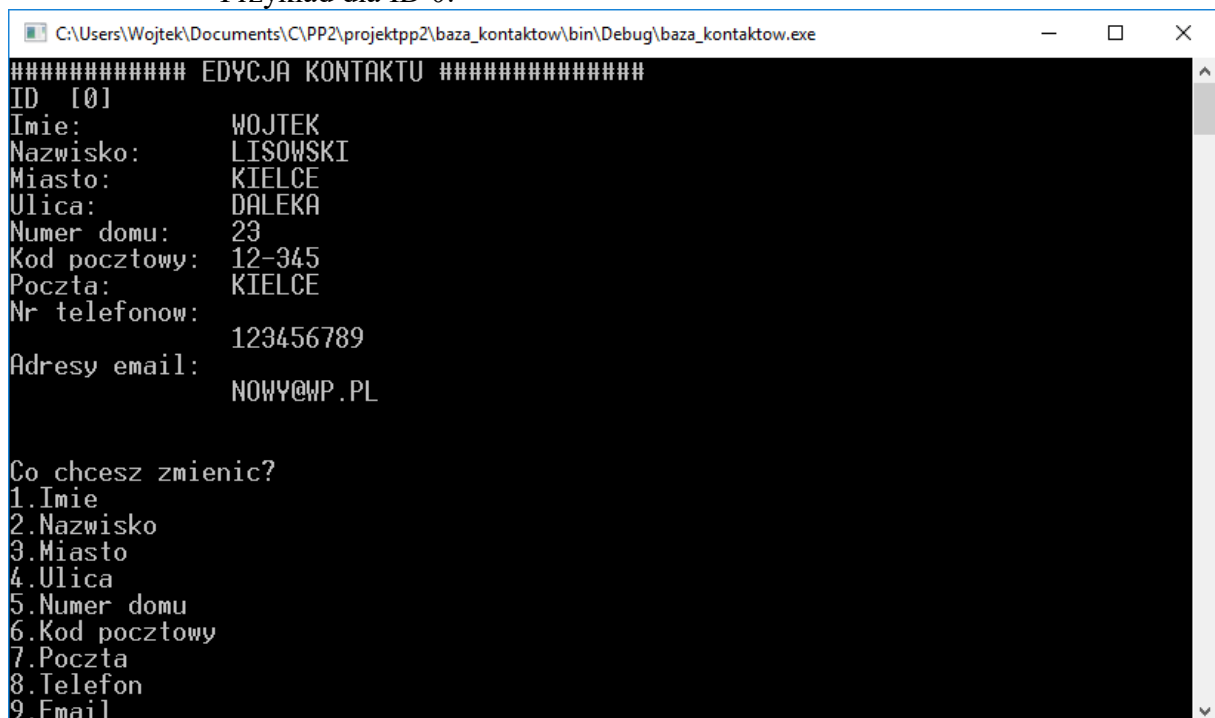
```
C:\Users\Wojtek\Documents\C\PP2\projektpp2\baza_kontaktow\bin\Debug\baza_kontaktow.exe
#####
ID [0]
Imie:      WOJTEK
Nazwisko:   LISOWSKI
Miasto:     KIELCE
Ulica:      DALEKA
Numer domu: 23
Kod pocztowy: 12-345
Poczta:     KIELCE
Numery telefonow:
            123456789
Adresy email:
            NOWY@WP.PL
#####
ID [1]
Imie:      WOJTEK
Nazwisko:   NOWAK
Miasto:     RADOM
Ulica:      BLISKA
Numer domu: 99
Kod pocztowy: 99-324
Poczta:     RADOM
Numery telefonow:
            987654321
            987654322
```

Aby przejść do menu głównego klikamy ENTER

4.3. Edycja kontaktu.

Jeśli chcemy edytować jakiegoś użytkownika wybieramy opcję nr „3” z menu głównego, następnie podajemy ID użytkownika, którego chcemy edytować. To pole też posiada walidację danych.

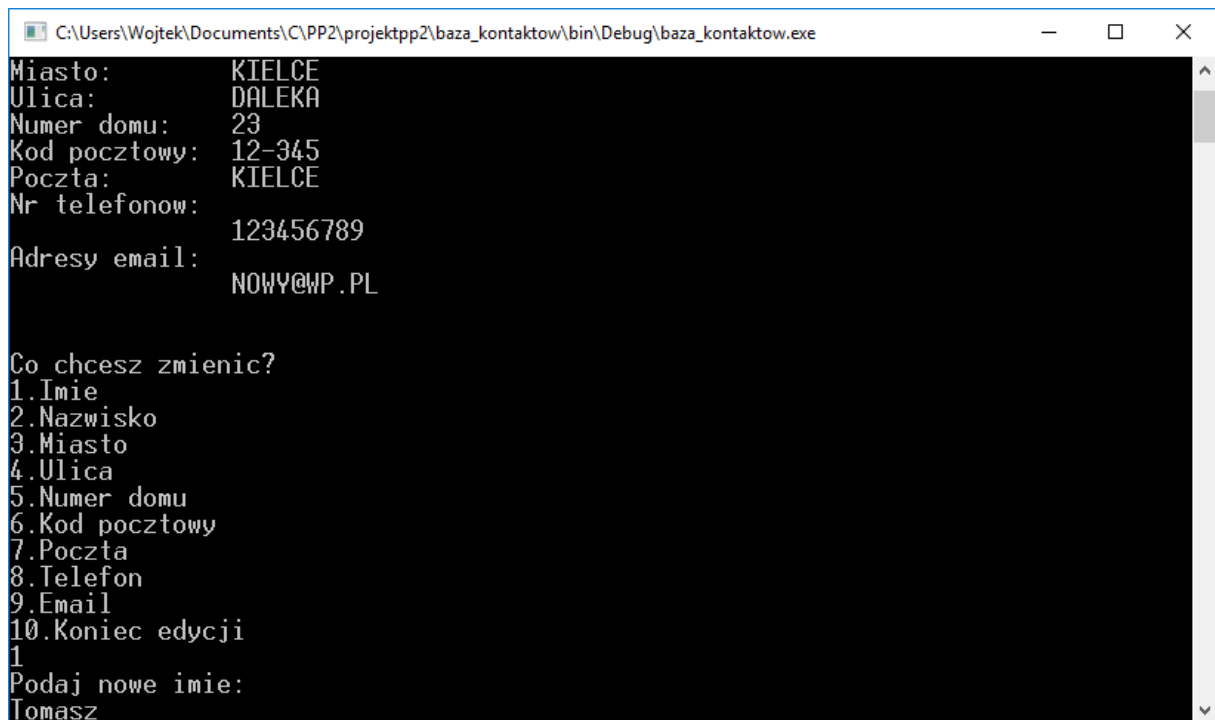
Przykład dla ID 0:



```
C:\Users\Wojtek\Documents\C\PP2\projektpp2\baza_kontaktow\bin\Debug\baza_kontaktow.exe
##### EDYCJA KONTAKTU #####
ID [0]
Imie:      WOJTEK
Nazwisko:   LISOWSKI
Miasto:     KIELCE
Ulica:      DALEKA
Numer domu: 23
Kod pocztowy: 12-345
Poczta:     KIELCE
Nr telefonow:
            123456789
Adresy email:
            NOWY@WP.PL

Co chcesz zmienic?
1.Imie
2.Nazwisko
3.Miasto
4.Ulica
5.Numer domu
6.Kod pocztowy
7.Poczta
8.Telefon
9.Email
```

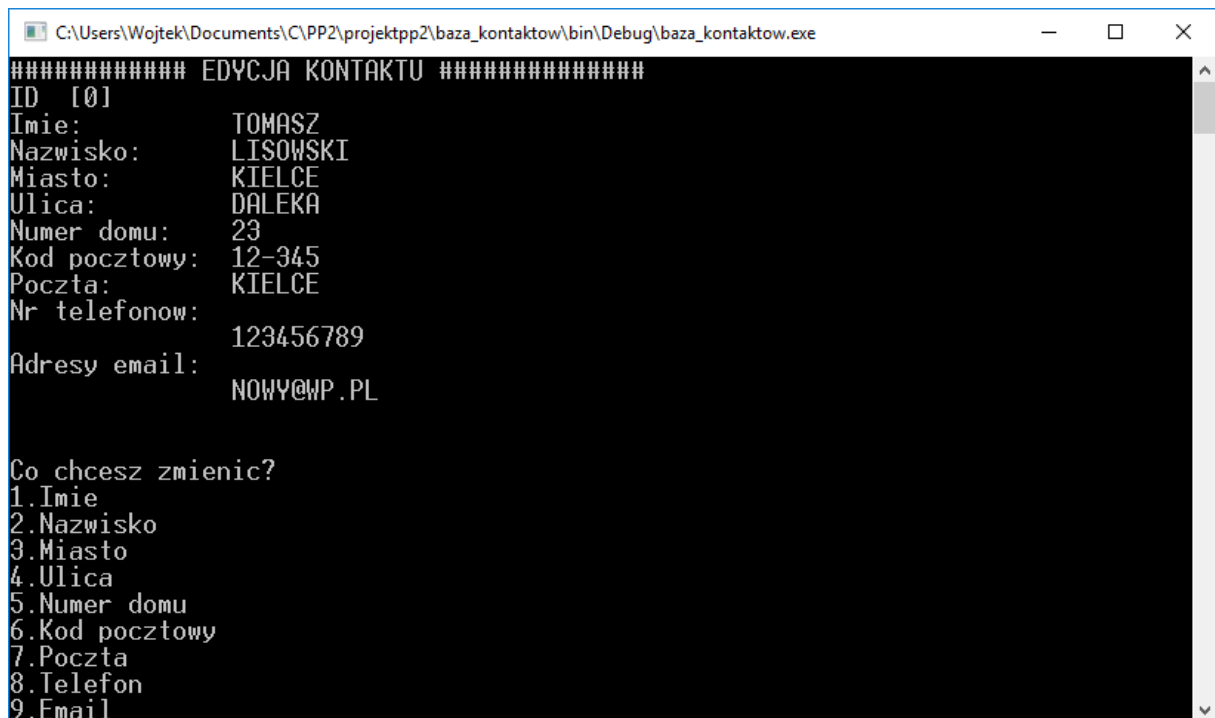
W tym menu możemy edytować każde pole z wyjątkiem ID. Przykładowe działanie dla pola „imię”:



```
C:\Users\Wojtek\Documents\C\PP2\projektp2\baza_kontaktow\bin\Debug\baza_kontaktow.exe
Miasto:      KIELCE
Ulica:      DALEKA
Numer domu:  23
Kod pocztowy: 12-345
Poczta:     KIELCE
Nr telefonow: 123456789
Adresy email: NOWY@WP.PL

Co chcesz zmienic?
1.Imie
2.Nazwisko
3.Miasto
4.Ulica
5.Numer domu
6.Kod pocztowy
7.Poczta
8.Telefon
9.Email
10.Koniec edycji
1
Podaj nowe imie:
Tomasz
```

Podaliśmy nowe imię „Tomasz”.



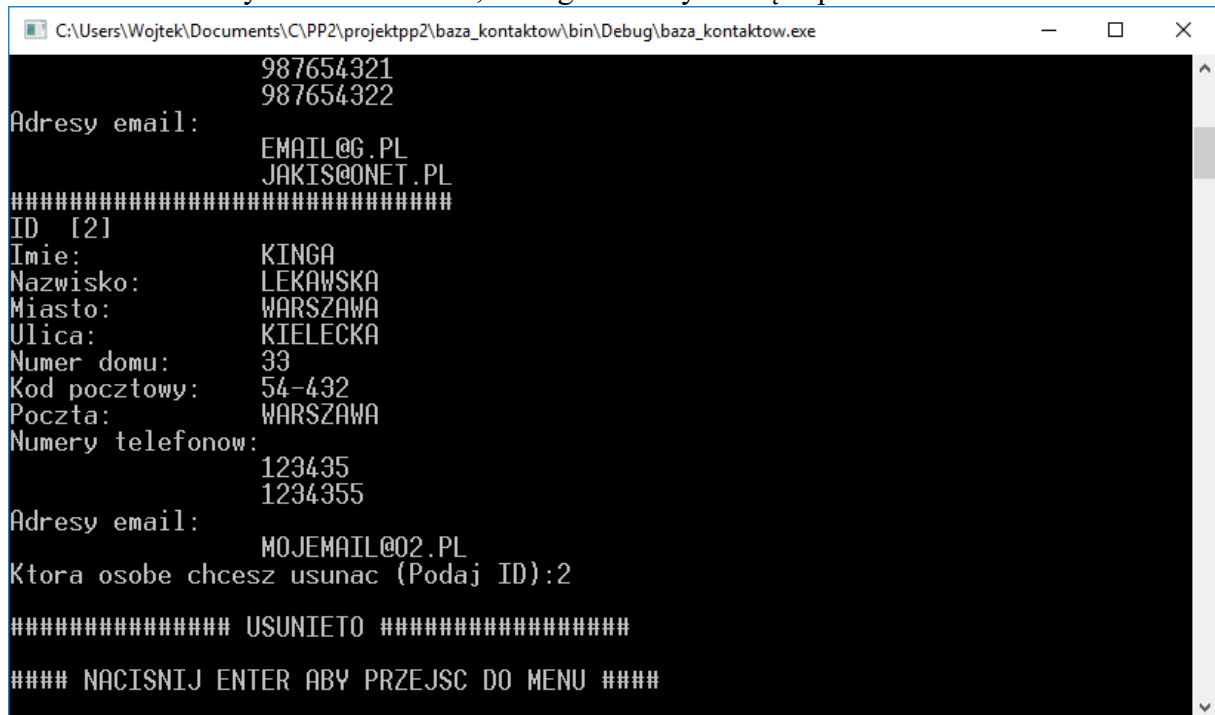
```
C:\Users\Wojtek\Documents\C\PP2\projektp2\baza_kontaktow\bin\Debug\baza_kontaktow.exe
##### EDYCJA KONTAKTU #####
ID [0]
Imie:      TOMASZ
Nazwisko:  LISOWSKI
Miasto:    KIELCE
Ulica:     DALEKA
Numer domu: 23
Kod pocztowy: 12-345
Poczta:    KIELCE
Nr telefonow: 123456789
Adresy email: NOWY@WP.PL

Co chcesz zmienic?
1.Imie
2.Nazwisko
3.Miasto
4.Ulica
5.Numer domu
6.Kod pocztowy
7.Poczta
8.Telefon
9.Email
```

Po potwierdzeniu klawiszem ENTER imię zostało zmienione z „WOJTEK” na „TOMASZ”. Resztę pól edytujemy analogicznie. Wszystkie pola posiadają taką samą walidację wprowadzanych danych jak przy dodawaniu nowego kontaktu.

4.4. Usuwanie kontaktu.

Aby usunąć kontakt trzeba wybrać opcję nr „2” w menu głównym. Następnie wybrać ID kontaktu, którego chcemy usunąć np.:



```
C:\Users\Wojtek\Documents\C\PP2\projektpp2\baza_kontaktow\bin\Debug\baza_kontaktow.exe

987654321
987654322
Adresy email:
EMAIL@G.PL
JAKIS@ONET.PL
#####
ID [2]
Imie: KINGA
Nazwisko: LEKAWSKA
Miasto: WARSZAWA
Ulica: KIELECKA
Numer domu: 33
Kod pocztowy: 54-432
Pocztą: WARSZAWA
Numery telefonow:
123435
1234355
Adresy email:
MOJEMAIL@02.PL
Ktora osobe chcesz usunac (Podaj ID):2
##### USUNIETO #####
#### NACISNIJ ENTER ABY PRZEJSC DO MENU ####
```

4.5. Wyszukiwanie kontaktu

Aby wyszukać kontakt wybieramy opcję nr „5” w menu głównym. Szukać można po 3 polach(ID, imię, nazwisko). W tym przykładzie będziemy szukać po imieniu, więc wybieramy opcję nr 1.



```
C:\Users\Wojtek\Documents\C\PP2\projektpp2\baza_kontaktow\bin\Debug\baza_kontaktow.exe

Po jakim polu chcesz wyszukac?
0 - ID
1 - IMIENIU
2 - NAZWISKU
WYBOR: 1
```



```
C:\Users\Wojtek\Documents\C\PP2\projektpp2\baza_kontaktow\bin\Debug\baza_kontaktow.exe
```

```
Podaj imie uzytkownika, ktorego chcesz wyswietlic:
wojtekd
Podaj imie uzytkownika, ktorego chcesz wyswietlic:
wojtek
#####
ID [1]
Imie:          WOJTEK
Nazwisko:      NOWAK
Miasto:        RADOM
Ulica:         BLISKA
Numer domu:    99
Kod pocztowy:  99-324
Poczta:        RADOM
Nr telefonow:  987654321
                987654322
Adresy email:  EMAIL@G.PL
                JAKIS@ONET.PL
Nacisnij ENTER by kontynuowac
```

4.6. Zapis do pliku

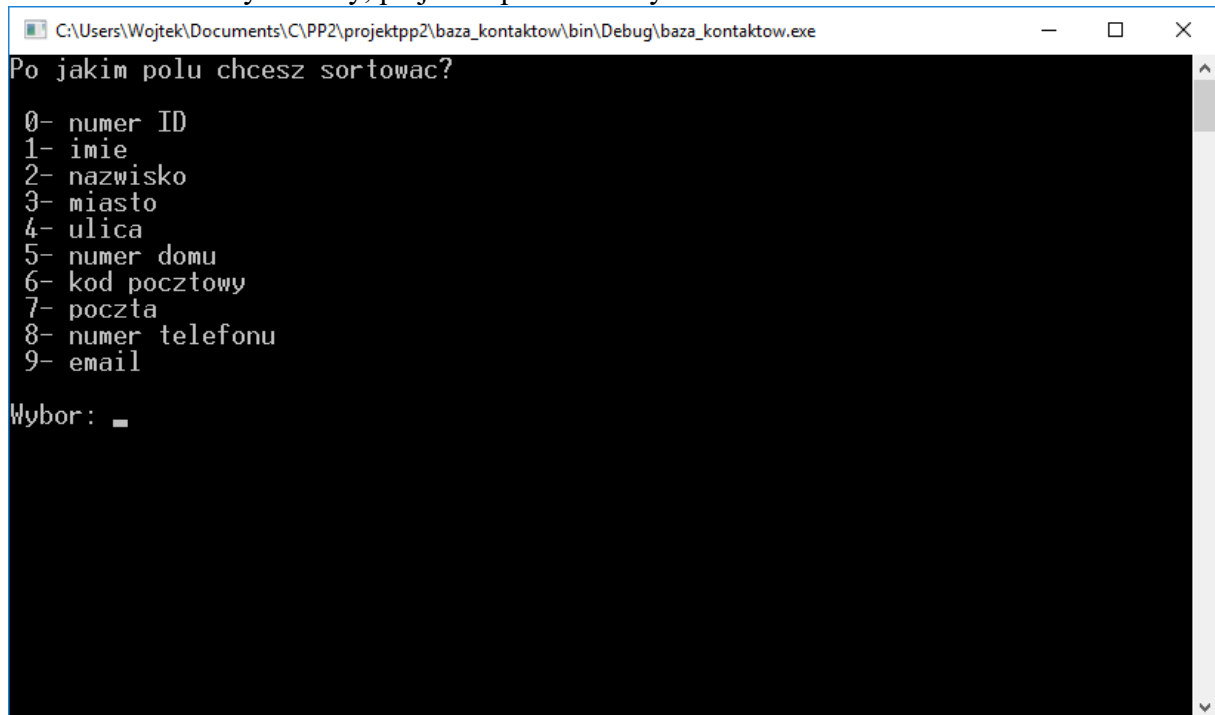
EMAIL.txt — Notatnik	
Plik	Edycja Format Widok
0	NOWY@WP.PL
1	EMAIL@G.PL
1	JAKIS@ONET.PL

USERS.txt — Notatnik	
Plik	Edycja
0	TOMASZ LISOWSKI KIELCE DALEKA 23 12-345 KIELCE 1 WOJTEK NOWAK RADOM BLISKA 99 99-324 RADOM

TELEPHONE.txt — Notatnik	
Plik	Edycja Format Widok P
0	123456789
1	987654321
1	987654322

4.7.Sortowanie

Aby posortować listę należy wybrać opcję nr „7”. W następnym menu wybieramy, po jakim polu chcemy sortować.



```
C:\Users\Wojtek\Documents\C\PP2\projektpp2\baza_kontaktow\bin\Debug\baza_kontaktow.exe
Po jakim polu chcesz sortowac?
0- numer ID
1- imie
2- nazwisko
3- miasto
4- ulica
5- numer domu
6- kod pocztowy
7- poczta
8- numer telefonu
9- email
Wybor: _
```

Dla przykładu posortujemy po imieniu, czyli wybieramy opcję nr „1”. Następnie wybieramy kierunek sortowania, w przykładzie posortujemy rosnąco, czyli opcja „0”.



```
C:\Users\Wojtek\Documents\C\PP2\projektpp2\baza_kontaktow\bin\Debug\baza_kontaktow.exe
Jak chcesz posortowac liste?
0- rosnaco
1- malejaco
Wybor: 0_
```

Wynik sortowania wyświetlamy opcją nr „4” w menu głównym.

```
C:\Users\Wojtek\Documents\C\PP2\projektp2\baza_kontaktow\bin\Debug\baza_kontaktow.exe

#####
ID [0]
Imie:      TOMASZ
Nazwisko:  LISOWSKI
Miasto:    KIELCE
Ulica:     DALEKA
Numer domu: 23
Kod pocztowy: 12-345
Poczta:    KIELCE
Numery telefonow: 123456789
Adresy email: NOWY@WP.PL
#####
ID [1]
Imie:      WOJTEK
Nazwisko:  NOWAK
Miasto:    RADOM
Ulica:     BLISKA
Numer domu: 99
Kod pocztowy: 99-324
Poczta:    RADOM
Numery telefonow: 987654321
                  987654322
```

„T” jest przed „W” w alfabecie, więc sortowanie działa poprawnie. Aby się upewnić, że sortowanie działa poprawie posortujemy jeszcze raz po polu „imię”, ale malejąco.

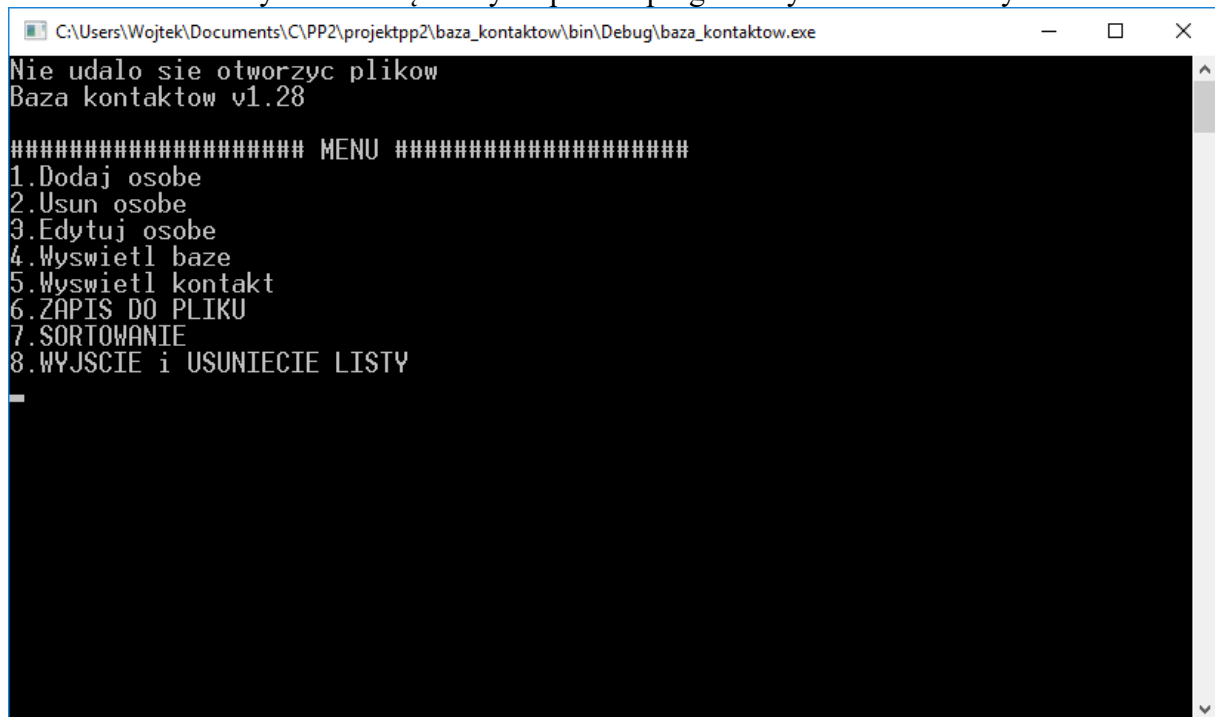
```
C:\Users\Wojtek\Documents\C\PP2\projektp2\baza_kontaktow\bin\Debug\baza_kontaktow.exe

#####
ID [1]
Imie:      WOJTEK
Nazwisko:  NOWAK
Miasto:    RADOM
Ulica:     BLISKA
Numer domu: 99
Kod pocztowy: 99-324
Poczta:    RADOM
Numery telefonow: 987654321
                  987654322
Adresy email: EMAIL@G.PL
                  JAKIS@ONET.PL
#####
ID [0]
Imie:      TOMASZ
Nazwisko:  LISOWSKI
Miasto:    KIELCE
Ulica:     DALEKA
Numer domu: 23
Kod pocztowy: 12-345
Poczta:    KIELCE
Numery telefonow:
```

Jak widać sortowanie działa poprawnie, ponieważ kontakty zostały zamienione miejscami.

4.8. Działanie programu, gdy nie udało się odczytać plików

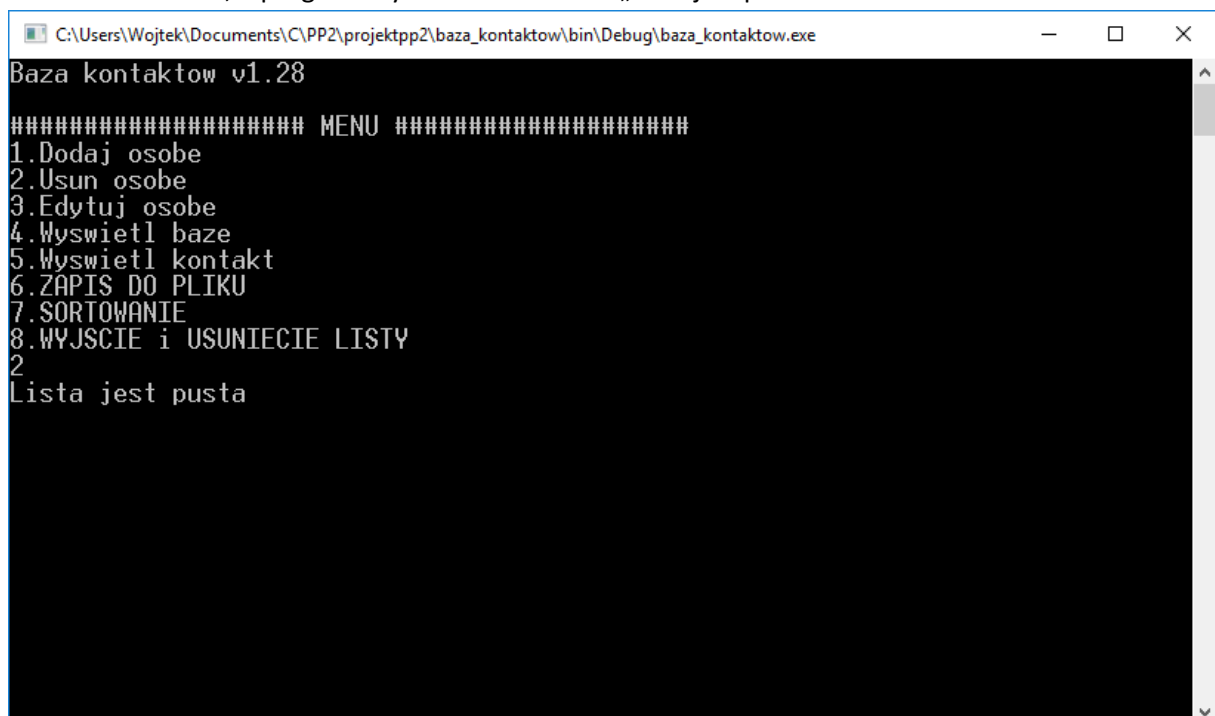
Gdy nie uda się odczytać plików program wyświetli stosowny komunikat:



```
C:\Users\Wojtek\Documents\C\PP2\projektpp2\baza_kontaktow\bin\Debug\baza_kontaktow.exe
Nie udało sie otworzyc plikow
Baza kontaktow v1.28

##### MENU #####
1.Dodaj osobe
2.Usun osobe
3.Edytuj osobe
4.Wyświetl baze
5.Wyświetl kontakt
6.ZAPIS DO PLIKU
7.SORTOWANIE
8.WYJSCIE i USUNIECIE LISTY
-
```

W takim przypadku dostępne będą tylko dwie opcje (1- Dodawanie osoby i 8-wyjście z programu) reszta nie zadziała, a program wyświetli komunikat „Lista jest pusta”:



```
C:\Users\Wojtek\Documents\C\PP2\projektpp2\baza_kontaktow\bin\Debug\baza_kontaktow.exe
Baza kontaktow v1.28

##### MENU #####
1.Dodaj osobe
2.Usun osobe
3.Edytuj osobe
4.Wyświetl baze
5.Wyświetl kontakt
6.ZAPIS DO PLIKU
7.SORTOWANIE
8.WYJSCIE i USUNIECIE LISTY
2
Lista jest pusta
```

5. Opis użytych algorytmów i najważniejszych fragmentów implementacji.

Program napisany jest na liście jednokierunkowej bez wartowników. Implementacja listy jest wzorowana na podstawie instrukcji laboratoryjnej nr 5 przygotowanej przez dr inż. Arkadiusza Chrobota i mgr inż. Pawła Piętę.

Kilka ważnych elementów programu:

Dodawanie numeru telefonu do listy:

```
41 struct telephone_nr * InsertTelNum(struct telephone_nr
*FrontTelList)
42 {
43     struct telephone_nr *new_node = (struct telephone_nr *)malloc
(sizeof (struct telephone_nr));
44     struct telephone_nr *wsk=FrontTelList;
45     unsigned long int NrTel;
46     while (wsk->next != NULL)
47     {
48         wsk = wsk->next;
49     }
50     if (NULL != new_node)
51     {
52         system("cls");
53         puts("#### DODAWANIE UZYTOWNIKA ####");
54         puts("Podaj numer telefonu:");
55         wsk->next = new_node;
56         while (scanf("%lu", &NrTel) != 1 || NrTel<1 ||
NrTel>999999999)
57         {
58             puts("Niepoprawny format, podaj numer tel jeszcze raz:");
59             fflush(stdin);
60         }
61         new_node->ID_tel=NumID;
62         new_node->tel_nr=NrTel;
63         new_node->next=NULL;
64     }
65     return FrontTelList;
66 }
```

Funkcja ta dodaje na koniec listy nowy numer telefonu.

Szczegółowy opis:

- W linii 43 przydzielany jest nowy obszar pamięci
- W linii 44 wskaźnikowi roboczemu przypisujemy adres pierwszego elementu listy
- W linii 45 deklarujemy zmienną, która będzie przechowywać numer telefonu podany przez użytkownika
- W liniach 46-49 przesuwamy roboczy wskaźnik na koniec listy
- W linii 50 sprawdzamy czy udało się przydzielić pamięć na nowy element
- W linii 52 czyścimy ekran
- W linii 55 dołączamy nowy element na koniec listy
- W linii 56 podajemy numer telefonu, warunek sprawdza czy podany nr nie jest ciągiem znaków i czy jest z przedziału od 1 do 999999999 jeśli któryś z

warunków się spełni to zostaną wykonane linie 58 i 59 po czym użytkownik będzie musiał podać nr jeszcze raz

- W linii 58 wyświetlany jest komunikat o błędzie
- W linii 59 czyszczony jest bufor
- W linii 61 zapisywane jest ID numeru telefonu
- W linii 62 zapisywany jest numer telefonu do listy
- W linii 63 wskaźnikowi na następny element przypisywana jest wartość NULL
- W linii 65 funkcja zwraca wskaźnik na pierwszy element listy

Funkcje dodające nowe adresy e-mail i nowych użytkowników do list działają na tej samej zasadzie tylko różnią się walidacją danych. Kilka funkcji sprawdzających poprawność danych zostanie omówiona poniżej.

Funkcja sprawdzająca poprawność kodu pocztowego:

```
359 int check_postal_code(char array[])
360 {
361     if(strlen(array)>6) return 0;
362     if(isdigit(array[0])==0) return 0;
363     if(isdigit(array[1])==0) return 0;
364     if((array[2])!='-') return 0;
365     if(isdigit(array[3])==0) return 0;
366     if(isdigit(array[4])==0) return 0;
367     if(isdigit(array[5])==0) return 0;
368     else
369         return 1;
370 }
```

- W linii 359 funkcja jako parametr przyjmuje tablicę znaków
- W linii 361 sprawdzana jest długość ciągu jeśli jest dłuższa od 6 to funkcja zwróci wartość 0
- W liniach 362-363 sprawdzamy czy w indexie 0 i 1 nie ma liczby
- W linii 364 sprawdzamy czy w tablicy o indexie 2 nie ma znaku „-”
- W liniach 365-367 sprawdzamy czy nie ma liczb w tablicy o indeksach 3-5
- W linii 369 funkcja zwróci wartość 1 jeśli żaden z poprzednich warunków nie został spełniony

Funkcja sprawdzająca poprawność podanego ciągu

```
int check_string(char array[])
379 {
380     int i,k=1;
381     for(i=0; i<strlen(array); ++i)
382     {
383         if(isalnum(array[i])!=0 && isdigit(array[i])==0)
384         {
385         }
386         else
387         {
388             k=0;
389         }
390     }
391     if(k==0) return 0;
```

```

392     return 1;
393 }

```

- W linii 383 pierwszy warunek sprawdza czy podany znak jest liczbą lub literą, a drugi sprawdza czy znak nie jest liczbą
- W linii 388 zmiennej k jest powypisywana wartość 0 gdy warunek nie zostanie spełniony
- W liniach 391-392 funkcja zwraca wartość 0 gdy k==0 lub 1 gdy k==1

Funkcja sprawdzająca czy użytkownik o podanym ID istnieje:

```

421 int user_exist(struct User_Node * FrontUserList, int data)
422 {
423     for (; NULL != FrontUserList; FrontUserList = FrontUserList ->
next)
424         if(FrontUserList->ID==data)
425         {
426             return 1;
427             break;
428         }
429     return 0;
430 }

```

- Funkcja, jako argumenty pobiera wskaźnik na pierwszy element w liście i ID podane przed użytkownika
- W linii 423 poruszamy się po liście
- Jeśli warunek z linii 424 zostanie spełniony to funkcja zwróci 1 i przerwie swoje działanie.

Procedura zapisująca dane do plików:

```

void save(FILE *file, FILE *file2, FILE *file3, struct User_Node *
FrontUserList, struct telephone_nr * FrontTelList, struct email_node
*FrontEmailList)
19 {
20     file=fopen("USERS.txt", "w");
21     file2=fopen("TELEPHONE.txt", "w");
22     file3=fopen("EMAIL.txt", "w");
23     if(file==NULL || file2==NULL || file3==NULL)
24     {
25         printf("Bład odczytu pliku");
26         exit(0);
27     }
28
29     for (; NULL != FrontUserList ; FrontUserList = FrontUserList ->
next )
30     {
31         fprintf (file, "\n%u\n%s\n%s\n%s\n%s\n%hu\n%s\n%s",
32                 FrontUserList->ID, FrontUserList->name,
FrontUserList->surname, FrontUserList -> city,
33                 FrontUserList->street, FrontUserList->house_number,
FrontUserList->postal_code,
34                 FrontUserList->post_office);
35     }
36     for (; NULL != FrontTelList ; FrontTelList = FrontTelList -> next
)
37     {

```

```

38         fprintf(file2, "\n%u\t", FrontTelList->ID_tel);
39         fprintf(file2, "%lu", FrontTelList->tel_nr);
40     }
41     for (; NULL != FrontEmailList ; FrontEmailList = FrontEmailList ->
next )
42     {
43         fprintf(file3, "\n%u\t", FrontEmailList->ID_email);
44         fprintf(file3, "%s", FrontEmailList->em);
45     }
46
47     if(fclose(file) || fclose(file2) || fclose(file3))
48     {
49         printf("Bład zamknięcia pliku");
50     }
51 }

```

- W liniach 20-22 otwierane są pliki do zapisu
- W linii 23 sprawdzane jest czy udało się utworzyć pliki
- W liniach 29-35 zapisywane są dane kontaktów do pliku USERS.txt
- W liniach 36-40 zapisywane są numery telefonów do pliku TELEPHONE.txt
- W liniach 41-45 zapisywane są adresy e-mail do pliku EMAIL.txt
- W liniach 47 sprawdzamy czy poprawnie zamknięto pliki

6. Podsumowanie:

Nie udało się zrealizować dwuosobowego projektu. Projekt napisała jedna osoba.
No i parę rzeczy można by poprawić :D