

Napisz program, który wczyta do tablicy struktur dane na podstawie pliku tekstowego i wyświetli je w konsoli. Rekordy w pliku zawierają nazwę kategorii, nazwę produktu i cenę produktu. Przygotuj strukturę `categories_t` do przechowywania nazw kategorii oraz produktów:

```
struct categories_t
{
    char* name;
    struct products_t* products;
    unsigned int count;
};
```

gdzie:

- `name` - przechowuje nazwę kategorii,
- `products` - przechowuje produkty w danej kategorii.
- `count` - przechowuje ilość produktów w danej kategorii.

Przygotuj strukturę `products_t` do przechowywania nazw produktów i cen:

```
struct products_t
{
    char* product_name;
    unsigned int price;
};
```

gdzie:

- `product_name` - przechowuje nazwę produktu,
- `price` - przechowuje cenę produktu.

Zaimplementuj następujące API:

```
int read_categories(const char* filename, struct categories_t*** categories);
void display_categories(struct categories_t** categories);
void free_categories(struct categories_t** categories);
```

Definicje wszystkich typów danych wraz z prototypami funkcji API należy dostarczyć w pliku `defs.h`.

```
int read_categories(const char* filename, struct categories_t*** categories);
```

Funkcja wczytuje dane z pliku tekstowego `filename` i dodaje je do tablicy struktur w kolejności, w jakiej pojawiają się w pliku. W tablicy struktur powinny znajdować się tylko unikalne dane, jeżeli wczytana z pliku nazwa kategorii występuje już w tablicy to funkcja powinna dodać nowy produkt wraz z ceną do produktów z danej kategorii.

Funkcja może przydzielić tylko tyle pamięci, ile jest potrzebnej do przechowania wszystkich danych.

Przykład dane w pliku:

```
X-wing|Luke's Legacy|220000¤
TIE Fighter|Vader's Fury|75000¤
TIE Fighter|Kylo's Shadow|80000¤
Millennium Falcon|Chewie's Faithful|1000000¤
V-wing|Wedge's Steadfast|165000¤
Imperial Star Destroyer|Palpatine's Wrath|14500000¤
X-wing|Red Five|220000¤
TIE Fighter|Scimitar Squadron|80000¤
Millennium Falcon|Hypernova Explorer|950000¤
V-wing|Cosmic Phoenix|170000¤
Imperial Star Destroyer|Resurgent-class Star Destroyer|14000000¤
X-wing|Rebel Starfighter|23000¤
TIE Fighter|First Order Special Forces TIE Fighter|8000¤
Millennium Falcon|Solo's Pride|950000¤
V-wing|Gold Leader's Revenge|165000¤
Imperial Star Destroyer|Imperial-II-class Star Destroyer|14500000¤
```

Wartość zwracana:

- `0` – w przypadku sukcesu,
- `1` – w przypadku przekazania do funkcji błędnych parametrów,
- `2` – w przypadku, kiedy plik wejściowy nie istnieje,
- `3` – w przypadku, kiedy dane w pliku były niepoprawne,
- `4` – w przypadku niepowodzenia alokacji pamięci.

```
void display_categories(struct categories_t** categories);
```

Funkcja wyświetla w konsoli zawartość tabeli struktur. Dane z tabeli powinny być wyświetlane według wzorca:

```
Kategoria:  
1. Produkt1 - cena  
2. Produkt2 - cena  
.  
. .  
N. ProduktN - cena
```

W przypadku przekazania błędnych danych funkcja nic nie robi.

```
void free_categories(struct categories_t** categories);
```

Funkcja zwalnia pamięć przydzieloną zarówno na tablicę struktur, jak i tablicę produktów. W przypadku przekazania błędnych danych funkcja nic nie robi.

Napisz program, który w oparciu o przygotowane funkcje wczyta z pliku dane, a następnie wyświetli je w konsoli. Jeżeli operacja otwarcia pliku się nie powiedzie, to program powinien wyświetlić komunikat `Couldn't open file` i zakończyć się z kodem błędu 4. Po wczytaniu pliku program powinien wyświetlić zawartość tablicy struktur w taki sam sposób, jak funkcja `display_categories`.

Jeżeli wystąpi błąd związany z alokacją pamięci, to program powinien wyświetlić komunikat `Failed to allocate memory` i zakończyć się z kodem błędu 8. Jeżeli dane w pliku będą uszkodzone, to program powinien wyświetlić komunikat `File corrupted` i zakończyć się z kodem błędu 6.

Przykład interakcji (błąd alokacji):

```
Failed to allocate memory
```

Przykład interakcji (nieistniejący plik):

```
Couldn't open file
```

Przykład interakcji (dane uszkodzone):

```
File corrupted
```

Przykład interakcji (sukces): [star_wars.txt](#)

Zawartość pliku star_wars.txt

```
X-wing|Lukes Legacy|220000
X-wing|Leias Hope|240000
X-wing|Hans Pride|230000
TIE Fighter|Vaders Fury|75000
TIE Fighter|Kylos Shadow|80000
TIE Fighter|Mauls Revenge|77000
Millennium Falcon|Chewies Faithful|10000000
Millennium Falcon|Landos Luck|9500000
Y-wing|Wedges Steadfast|165000
Y-wing|Biggss Valor|170000
Imperial Star Destroyer|Palpatines Wrath|145000000
Imperial Star Destroyer|Thrawns Ascendancy|140000000
Imperial Star Destroyer|Vaders Fist|143000000
X-wing|Red Five|220000
X-wing|Blue Leader|240000
X-wing|Gold Squadron|230000
TIE Fighter|Black Squadron|75000
TIE Fighter|Scimitar Squadron|80000
TIE Fighter|Shadow Wing|77000
Millennium Falcon|Stardust Voyager|10000000
Millennium Falcon|Hypernova Explorer|9500000
Y-wing|Starblaze Enforcer|165000
Y-wing|Cosmic Phoenix|170000
Imperial Star Destroyer|Eclipse|145000000
Imperial Star Destroyer|Resurgent-class Star Destroyer|140000000
Imperial Star Destroyer|Victory-class Star Destroyer|143000000
X-wing|Rogue Leader|220000
```

X-wing:^d

1. Lukes Legacy - 220000^d
2. Leias Hope - 240000^d
3. Hans Pride - 230000^d
4. Red Five - 220000^d
5. Blue Leader - 240000^d
6. Gold Squadron - 230000^d
7. Rogue Leader - 220000^d

TIE Fighter:^d

1. Vaders Fury - 75000^d
2. Kylos Shadow - 80000^d
3. Mauls Revenge - 77000^d
4. Black Squadron - 75000^d
5. Scimitar Squadron - 80000^d
6. Shadow Wing - 77000^d

Millennium Falcon:^d

1. Chewies Faithful - 10000000^d
2. Landos Luck - 9500000^d
3. Stardust Voyager - 10000000^d
4. Hypernova Explorer - 9500000^d

Y-wing:^d

1. Wedges Steadfast - 165000^d
 2. Biggss Valor - 170000^d
 3. Starblaze Enforcer - 165000^d
 4. Cosmic Phoenix - 170000^d
- Imperial Star Destroyer:^d
1. Palpatines Wrath - 145000000^d
 2. Thrawns Ascendancy - 140000000^d
 3. Vaders Fist - 143000000^d
 4. Eclipse - 145000000^d
 5. Resurgent-class Star Destroyer - 140000000^d
 6. Victory-class Star Destroyer - 143000000^d

Uwagi

- W programie nie wolno używać operatora `[]`.
- W programie można zadeklarować pomocnicze tablice do wczytywania danych z pliku, deklaracja tablic **musi** wyglądać następująco:

```
char categorie_name[41], product_name[61];
```

lub:

```
char input[110];
```