Programowanie Strukturalne

12-12-2022

Niech dana będzie struktura:

```
typedef struct vector
{
    int size;
    int capacity;
    int * array;
} vector;
```

gdzie `size` wskazuje ilość przechowywanych *danych*, a `capacity` pojemność tablicy `array`. Napisz poniższe funkcje, tak aby zachowywały poprawność danych w strukturze `vector`, tzn. $0 \leqslant$ `size` $\leqslant$ `capacity` oraz `array` wskazywał na `capacity`-elementową tablicę (z wyjątkiem funkcji `init`, która wymusza poprawność danych). Napisz funkcje, które:

- `void init(vector *V)` inicjalizuje `V`, tak aby `size` $= 0$, `capacity` $= 1$,
- `void print(vector V)` drukuje *dane* wraz z opisem (polami `size`, `capacity`),
- `void relocate(vector *V)` podwaja parametr `capacity` zachowując *dane*,
- `void push_back(vector *V,int i)` dodaje `i` jako dodatkową informację na końcu *danych*,
- `void pop_back(vector *V)` usuwa ostatnią przechowywaną *daną* o ile istnieje,
- `void insert(vector *V,int at,int i)` jeśli $0 \leqslant$ `at` $<$ `size`, to dodaje `i` jako dodatkową informację na pozycje `at`, przesuwając oryginalne dane, począwszy od pozycji `at` o jedną pozycję w prawo,
- `void sort(vector *V)` sortuje *dane* w porządku niemalejącym $\leqslant$,
- `void reverse(vector *V)` odwraca kolejność *danych*,
- `void shrink_to_fit(vector *V)` zmniejsza `capacity` do większej z liczb `size` i 1,
- `void reserve(vector *V, int n)` zwiększa `capacity` do większej z liczb `size` i `n`,
- `void vector2file(vector V,char * file)` zapisuje *dane* do pliku `file`,
- `void file2vector(char * file,vector *V)` odczytuje *dane* z pliku `file` (zapisanych za pomocą funkcji `vector2file`) i umieszcza je w `V`.