

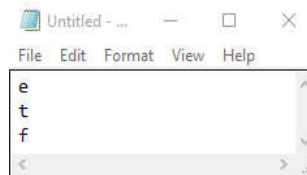
## PROGRAMIRANJE II (20.06.2017.)

❶ (20 bodova) Napisati funkciju:

```
void pisi (char *string, char *nazivFajla, char *komparator)
```

Koja rekurzivno upisuje po jedan karakter stringa (sa lijevo na desno) za svaki rekurzivni poziv funkcije u novi red tekstualne datoteke čiji je naziv drugi argument funkcije. Pri tome karakter se upisuje samo u slučaju ako se isti ne nalazi u stringu komparator. Za pretragu koristiti sekvencijalno pretraživanje sa stražom.

Za string *etf* (u slučaju da e, t i f nisu pronađeni u stringu komparator) upis bi u tekstualnoj datoteci bio:



Napisati program u kojem se string za upis i naziv datoteke preuzimaju kao argumenti komandne linije te se poziva prethodno implementirana funkcija. Nije dozvoljena upotreba statičkih, globalnih promjenljivih i statičkih nizova.

❷ (30 bodova) Napisati funkciju koja sortira niz pokazivača na funkcije. Funkcije se porede na osnovu vrijednosti koju funkcija vrati za zadati realan broj x. Za sortiranje je potrebno koristiti *insert-sort* algoritam. Prototip funkcije je:

```
void f_insert_sort(double (*f[])(double), int n, double x);
```

Napisati funkciju koja za svaku funkciju iz zadatog niza funkcija, ispisuje vrijednost funkcije za zadati realan broj x. Prototip funkcije je:

```
void f_print(double (*f[])(double), int n, double x);
```

Napisati glavni program u kojem treba:

- definisati niz pokazivača na funkcije iz standardne biblioteke *sin*, *cos*, *exp* i *sqrt*.
- iz binarne datoteke čiji je naziv prvi argument komandne linije čitati jedan po jedan binarno upisan realan broj tipa *double*, i pri tome, za svaku pročitane vrijednost, sortirati (korištenjem funkcije *f\_insert\_sort*) kreirani niz funkcija na osnovu pročitane vrijednosti i ispisati vrijednosti sortiranog niza funkcija za pročitane vrijednosti.

❸ (20 bodova) Neka su definisani tipovi:

```
typedef struct node {  
    void *data;  
    struct node *left, *right;  
} NODE;  
  
typedef struct bst {  
    NODE *root; // korijen stabla  
    int (*cmp)(const void *a, const void *b); // funkcija za poredjenje dva elementa  
} BST;
```

koji predstavljaju čvor stabla binarne pretrage sa podacima proizvoljnog tipa i stablo binarne pretrage, respektivno.

Napisati funkciju koja dodaje novi čvor u stablo binarne pretrage. Prototip funkcije je:

```
void add_node(BST *bst, void *data);
```

Napisati funkciju koja obilazi stablo u *inorder* poretku. Prototip funkcije je:

```
void inorder(BST *bst, void (*process)(NODE *n))
```

gdje je *process* funkcija koja vrši obradu informacionog sadržaja pri posjeti čvora.

**Napomena:** Dodavanje novog čvora i obilazak stabla je potrebno realizovati rekurzivno.

❹ (30 bodova) Definirati strukture steka i grafa tako da je informacioni sadržaj čvora steka ulančana reprezentacija grafa. Informacioni sadržaj čvora grafa je broj sa razlomljenim dijelom. Napisati funkciju koja preuzima sadržaj sa steka te vraća čvor iz grafova, koji su bili na steku, čiji je informacioni sadržaj najbliži nuli po apsolutnoj vrijednosti te je stepen tog čvora jednak ili veći od *n* (*n* je argument funkcije). U glavnom programu kreirati dva grafa sa pripadajućim čvorovima i granama po izboru, dodati grafove na stek te pozvati funkciju.