ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΕΣ ΤΕΧΝΙΚΕΣ

http://www.softlab.ntua.gr/~nickie/Courses/progtech/

Διδάσκοντες: Γιάννης Μαΐστρος (maistros@cs.ntua.gr)
Στάθης Ζάχος (zachos@cs.ntua.gr)
Νίκος Παπασπύρου (nickie@softlab.ntua.gr)

Διαφάνειες παρουσίασης #6 (β)

- ✓Ταξινομημένες λίστες
- ✓Παράμετροι του προγράμματος
- ✓Συναρτήσεις εισόδου-εξόδου
- ✓Συναρτήσεις διαχείρισης αρχείων

Ταξινομημένες λίστες (i)

- Τύπος slist

```
typedef ListNode * slist;
```

- Άδεια λίστα

```
const slist slistEmpty = NULL;
```

- Εισαγωγή στοιχείου

```
void slistInsert (slist * lp, int t)
{
    ListNode * n = (ListNode *)
       malloc(sizeof(ListNode));

    if (n == NULL) {
       printf("Out of memory\n");
       exit(1);
    }
```

Ταξινομημένες λίστες (ii)

- Εισαγωγή στοιχείου (συνέχεια)

```
    n->data = t;

    while (*lp != NULL && (*lp)->data < t)
        lp = &((*lp)->next);

    n->next = *lp;
    *lp = n;
}
```

Ταξινομημένες λίστες (iii)

- Αφαίρεση στοιχείου

```
void slistRemove (slist * lp, int t)
{
    ListNode * n;

    while (*lp != NULL && (*lp)->data < t)
        lp = &((*lp)->next);

    if (*lp == NULL) {
        printf("%d was not found\n", t);
        exit(1);
    }

    n = *lp;
    *lp = (*lp)->next;
    free(n);
}
```

Παράμετροι του προγράμματος (i)

- Επικεφαλίδα του προγράμματος

```
int main (int argc, char * argv[]);
```

- Παράμετροι
  - **`argc`** ο αριθμός των παραμέτρων
  - **`argv[i]`** η i-οστή παράμετρος
  - **`argv[0]`** το όνομα του προγράμματος
- Αποτέλεσμα
  - ακέραιος αριθμός που επιστρέφεται στο λειτουργικό σύστημα
  - συνήθως 0 για επιτυχή τερματισμό

Παράμετροι του προγράμματος (ii)

- Παράδειγμα

```
int main (int argc, char * argv[])
{
   int i;

   printf("Program %s called with "
          "%d parameters:\n", argv[0],
          argc-1);
   for (i = 1; i < argc; i++)
      printf(" %s", argv[i]);
   printf("\nand will return 0\n");

   return 0;
}
```

Συναρτήσεις εισόδου-εξόδου (i)

- Βασικές συναρτήσεις εισόδου-εξόδου

```
int printf (const char * format, ...);
int scanf (const char * format, ...);
```

- Ειδικοί χαρακτήρες στο format
  - Ακέραιοι αριθμοί

%d	στο δεκαδικό σύστημα
%u	χωρίς πρόσημο στο δεκαδικό σύστημα
%o	χωρίς πρόσημο στο οκταδικό σύστημα
%x	χωρίς πρόσημο στο δεκαεξαδικό σύστημα

Συναρτήσεις εισόδου-εξόδου (ii)

- Ειδικοί χαρακτήρες στο format
  - Αριθμοί κινητής υποδιαστολής

    %f σε μορφή: **[-]ddd.dddddd**

    %e σε μορφή: **[-]ddd.dddddd e [+/-]ddd**

    %g σε μορφή **%f** ή **%e**
  - Άλλοι τύποι

    %c χαρακτήρες

    %s συμβολοσειρές

    %p δείκτες

Συναρτήσεις εισόδου-εξόδου (iii)

- Παραλλαγές στο format
  - Μέγεθος αριθμών

%h	αριθμοί **short**	π.χ. **%hd**, **%hx**
%l	αριθμοί **long** ή **double**	π.χ. **%ld**, **%lf**
%L	αριθμοί **long double**	π.χ. **%Lf**

  - Μήκος αποτελέσματος

%8d	αριθμός σε μήκος 8 χαρακτήρων
%20s	συμβολοσειρά σε μήκος 20 χαρακτήρων
%+8d	αριθμός σε μήκος 8 χαρακτήρων με **+**
%08d	αριθμός σε μήκος 8 χαρακτήρων, τα πρώτα **0**
%-8d	όπως το **%8d** με στοίχιση αριστερά

Συναρτήσεις εισόδου-εξόδου (iv)

- Είσοδος-έξοδος χαρακτήρων

```
int putchar (int c);
int getchar ();
```

- Είσοδος-έξοδος συμβολοσειρών

```
int puts (const char * s);
char * gets (char * s);
```

- Έλεγχος τέλους δεδομένων

```
int eof ();
```

  - Η σταθερά **`EOF`** παριστάνει το τέλος των δεδομένων και έχει τύπο **`int`**.

Παράδειγμα

- Αντιγραφή δεδομένων
  - οι χαρακτήρες που διαβάζονται εκτυπώνονται, μέχρι να παρουσιαστεί τέλος δεδομένων

```
void main ()
{
   int c;

   while ((c = getchar()) != EOF)
      putchar(c);
}
```

Συναρτήσεις διαχείρισης αρχείων (i)

- Τύπος αρχείου

```
FILE * fp;
```

- Άνοιγμα αρχείων

```
FILE * fopen (const char * filename,
              const char * mode);
```

  - Παράμετρος mode

r	ανάγνωση (read)
w	εγγραφή (write)
a	προσθήκη (append)
t	κείμενο (text)
b	δυαδικά δεδομένα (binary)

Συναρτήσεις διαχείρισης αρχείων (ii)

- Κλείσιμο αρχείων

```
int fclose (FILE * fp);
```

- Είσοδος-έξοδος χαρακτήρων

```
int fputc (int c, FILE * fp);
int fgetc (FILE * fp);
```

- Είσοδος-έξοδος συμβολοσειρών

```
int fputs (const char * s, FILE * fp);
char * fgets (char * s, int n,
              FILE * fp);
```

Συναρτήσεις διαχείρισης αρχείων (iii)

- Βασικές συναρτήσεις εισόδου-εξόδου

```
int fprintf (FILE * fp,
             const char * format, ...);
int fscanf (FILE * fp,
             const char * format, ...);
```

- Έλεγχος τέλους αρχείου

```
int feof (FILE * fp);
```

Συναρτήσεις διαχείρισης αρχείων (iv)

- Είσοδος-έξοδος πολλών δεδομένων

```
size_t fwrite (const void * p,
  size_t size, size_t num, FILE * fp);
size_t fread (void * p,
  size_t size, size_t num, FILE * fp);
```

  - Ο ακέραιος τύπος **size_t** χρησιμοποιείται για τη μέτρηση χώρου μνήμης σε bytes.

Παράδειγμα (i)

- Αντιγραφή δυαδικών αρχείων

```
int main (int argc, char * argv[])
{
    FILE * fin, * fout;
    unsigned char buffer[1000];
    size_t count;

    fin = fopen(argv[1], "rb");
    if (fin == NULL)
        return 1;

    fout = fopen(argv[2], "wb");
    if (fout == NULL)
        return 2;
```

Παράδειγμα (ii)

- (συνεχίζεται)

```
    while (!feof(fin)) {
        count = fread(buffer, 1,
                      1000, fin);
        fwrite(buffer, 1, count, fout);
    }

    fclose(fin);
    fclose(fout);

    return 0;
}
```