Laborator 6 – Limbaje de Programare

1 Recapitulare şi exemple – Şiruri de caractere

Să ne amintim câteva exemple cu şiruri de caractere de la curs:

1.1 Citirea şi tipărirea şirurilor de caractere:

1. Citirea unui şir caracter cu caracter:

```
void readString(char a[], int n){
//citim un sir de maxim n caractere

  int i = 0;
  int c = getchar();

  while(c != EOF && i < n){
     a[i] = c;  //punem caract. citit in tablou
     i++;   //avansam indicele tabloului
     c = getchar();  //citim caract. urmator
  }

  a[i] = '\0';
 //marcam sfarsitul sirului
}
```

2. Citirea şi tipărirea formatată a şirurilor:

```
char sir[20];

//citire:
scanf ("%19s", &sir);  //adauga '\0' la sf.
scanf ("%19s", sir);   //echivalenta cu &sir

//tiparire
printf ("%s", sir);
```

1.2 Implementarea unor funcţii din *string.h*:

1. Lungimea unui şir de caractere (se numără caracterele până la '\0'):

```
size_t strlen(const char *s) {
// lungimea sirului, pana la '\0'

// size_t (stddef.h): tip pt. dimensiuni pozitive
// (unsigned sau long unsigned)

  int i=0;
  while (s[i] != '\0')
      i++;  // avanseaza pana intalneste '\0'

   return i; // nr. de caract.; '\0' nu e numarat
}
```

2. Compararea a două şiruri (se parcurg şirurile cu acelaşi indice cât timp caracterele de pe poziţia curentšunt egale, apoi se returnează diferenţa primelor caractere diferite):

```
int strcmp (const char *s1, const char *s2) {
// compara 2 siruri
  int i=0;

  while (s1[i] == s2[i] && s1[i] != '\0')
     i++; // egale dar s1[i] nu e '\0'

  return s1[i] - s2[i];

  // < 0 pt. s1 < s2
  // > 0 pt. s1 > s2
  // == 0 daca s1, s2 egale
}
```

3. Copierea unui şir într-un alt şir:

```
char *strcpy(char *dest, const char *src) {
 // copiaza src in dest

  int i=0;
  while (src[i]!='\0'){
  // copiaza pana la '\0'
     dest[i] = src[i];
     i++;
   }
  // trebuie sa avem loc in sirul dest !!!

  dest[i] = '\0';

  return dest; // returneaza dest prin conventie
}
```

4. Concatenarea a două şiruri (se parcurge întâi primul şir până la '\0', apoi se copiaza în continuare conţinutul celui de-al doilea şir, caracter cu caracter):

```
char *strcat(char *dest, const char *src) {
 // concateneaza src la dest
 // trebuie sa avem loc in coada lui dest !!!

  int i=0, j=0;
  while (dest[i] != '\0')
    ++i;

  while (src[j] != '\0'){
    dest[i] = src[j];
    ++i; ++j;
  }
  dest[i] = '\0';
  return dest;
}
```

5. Găsirea unui caracter într-un sir:

```
char *strchr(const char *s, char c) {
  //cauta caracterul c in sirul s
  int i=0;
```

```
  while (s[i] != '\0'){
    if (s[i] == c)
       return &s[i];
       //returneaza adresa caract. in sir
    i++;
  }

  return NULL;
  //returneaza NULL daca nu a fost gasit
}
```

1.3 Exemple de utilizare a unor funcţii din *string.h*

1. Găsirea indicelui şirului maxim (la compararea cu *strcmp*) dintr-un tablou de n şiruri de caractere:

```
int indiceMax(char *tab[], int n){
//max unui tablou de siruri de caractere

  int i=0, index=0;

  for(i=0;i<n;i++){
     if(strcmp(tab[i], tab[index]) > 0)
     //daca sirul tab[i] > sirul tab[index]
         index=i;
   }

   return index;
}
```

2. Concatenarea tuturor şirurilor dintr-un tablou de n şiruri, intr-un şir de caractere primit ca parametru (în limita spaţiului disponibil în şirul rezultat, spaţiu indicat de parametrul *rn*):

```
void all(char *tab[], int n, char *rez, int rn){
//concateneaza elementele unui tablou de siruri

 int i=0;
 if(strlen(tab[0]) < rn)  //daca e loc in rez
     strcpy(rez, tab[0]); //copiem primul sir

 for(i=1;i<n;i++)
   if(strlen(rez) + strlen(tab[i]) + 2) < rn){
    //daca mai avem loc in sirul rez

      strcat(rez, ", "); //cu virgula intre
      strcat(rez, tab[i]);
   }
}
```

2 Lucrarea 6

Să se implementeze următoarele funcţii cu şiruri de caractere, să se apeleze în **main** şi să se tipărească la ieşire rezultatele obţinute.

- Scrieţi o funcţie care adaugă un caracter oarecare, primit ca parametru, la sfârşitul unui şir primit ca parametru.

- Scrieţi o funcţie care returnează adevărat dacă într-un şir de caractere primit ca parametru, caracterele se află toate în ordine crescătoare, respectiv fals în caz contrar.
- Scrieţi o funcţie care ia ca parametri două şiruri de caractere şi returnează adevărat dacă toate caracterele din primul şir se regăsesc în cel de al doilea (ordinea nu contează).
- Scrieţi o funcţie care intercalează două şiruri de caractere într-un al treilea şir (acesta va conţine un caracter din primul şir, apoi unul din al doilea, apoi următorul caracter din primul şir, următorul caracter din al doilea, etc.); toate cele trei şiruri se primesc ca parametri.
- Scrieţi o funcţie care inversează conţinutul unui şir primit ca parametru; rezultatul va fi scris într-un al doilea şir, primit la rândul lui ca parametru.
- Scrieţi o funcţie care înlocuieşte toate apariţiile unui subşir într-un şir de caractere cu alt subşir, toate cele trei şiruri se primesc ca parametri.