|  |  |  |
| --- | --- | --- |
| FUNCȚIA | EFECT | EXEMPLE |
| strlen ( s1, s2 ) | calculează lungimea șirului | strlen(“azi e soare”) = 11 |
| strchr ( s1, c ) | caută un caracter într-un șir de caractere | strchr( “alarma”, ‘r’) = “rma” |
| strrchr ( s1, c ) | caută un caracter într-un șir de caractere | strrchr( “alarma”, ‘a’) = “arma” |
| strstr (s1, s2 ) | caută un subșir într-un șir | strstr(“tastatura sta pe masa”, “sta”) = “statura sta pe masa” |
| strcpy (s1, s2 ) | copiază un șir în alt șir | aux[101]=””;  strcpy(aux, “creion”) --> aux=”creion”; |
| strncpy (s1, s2, n ) | copiază primele n caractere din șirul 2 în șirul 1 | aux[101]=””;  strcpy(aux, “creion”, 3)  **aux[3]=’\0’** --> aux=”creion”; |
| strcat (s1, s2 ) | lipește șirul 2 la sfârșitul șirului 1 | aux[101]=””;  strcat(aux, “cu noroc”);  strcat(aux, “moara”); --> aux=”cu norocmoara”; |
| strncat (s1, s2, n ) | lipește primele n caractere din șirul 2 la sfârșitul șirului 1 | aux[101]=””;  strcat(aux, “moara”, 3);  aux[3]=’\0’ --> aux=”moa”; |
| strcmp (s1, s2 ) | compară alfabetic și returnează | strcmp(“ana”, “ana”)=0;  strcmp(“zidar”, “ziua”)<0;  strcmp(“ana”, “Ana”)>0; |
| stricmp (s1, s2 ) | compară alfabetic fără a face diferență între literele mici și / sau mari | stricmp(“ana”, “Ana”)=0; |
| strlwr (s1 ) | transformă șirul în majuscule | strlwr(“sOaRe”)=soare; |
| strupr (s1 ) | transformă șirul în minuscule | strupr(“sOaRe”)=SOARE; |
| isupper ( c ) | verifică dacă e majusculă | c=ce era; cc=ce urmeaza;  while(p=strstr(text, c))  { strcpy(aux, p+strlen(c));  strcpy(p, cc);  strcat(text, aux);  } cout<<text<<endl; |
| islower ( c ) | verifică dacă e minusculă |
| isdigit ( c ) | verifică dacă e cifră |
| isalpha ( c ) | verifică dacă e literă sau cifră |
| strtok ( s1, delim ) | elimină din șir toate caracterele din delimitator | char text[2001], sep[]=",.!?; :";  char \*p;  p=strtok(text, sep);  while(p)  {  //prelucrăm p care este ENTITATEA identificată în text  p=strtok(NULL, sep);  } |