Computer Programming 1 Lab

2020-11-26

Outline

- C Characters & Strings
  - Libraries
  - Functions
- Exercise

String

The string is a sequence of characters and ended with ‘\0’

Declaration

- String literal
  - “Hello world!” **(Not ‘Hello world!’)**
- Char array
  - char string[17]; **(include ‘\0’)**
- Char pointer
  - char* string;

String & Character -- Initialization

> We can put the literal string in read-only memory and copy the string to newly allocated memory on the stack.

```
char user[] = "John";
char user2[] = {'J', 'o', 'h', 'n'};
char *userPtr = "John";
// same as const char *userPtr = "John";
// (Invalid) user = "John";
```

Libraries and Functions

- Character-Handling Library: **<ctype.h>**
  - character-handling functions
- General Utilities Library: **<stdlib.h>**
  - string-conversion functions
- Standard Input/Output Library: **<stdio.h>**
  - string & character input/output functions
- String Handling Library: **<string.h>**
  - string-processing functions

Character Handling Library <ctype.h>

Prototype	Function Description
int isalpha(int ch);	Check if character is alphabetic
int ispunct(int ch);	Check if character is a punctuation character
int isdigit(int ch);	Check if character is decimal digit
int toupper (int c);	Convert lowercase letter to uppercase

```c
#include <stdio.h>
#include <ctype.h>

int main(void) {
  int i=0;
  char str[]="C++";
  while (str[i])
  {
    if (isalpha(str[i])) printf ("character %c is alphabetic\n",str[i]);
    else printf ("character %c is not alphabetic\n",str[i]);
    i++;
  }
  return 0;
  /*
  Output:
    character C is alphabetic
    character + is not alphabetic
    character + is not alphabetic
  */
}
```

```c
#include <stdio.h>
#include <ctype.h>
int main ()
{
  int i=0;
  char str[]="Test String.\n";
  char c;
  while (str[i])
  {
    c=str[i];
    putchar (toupper(c));
    i++;
  }
  return 0;
  /*
  Output:
    TEST STRING.
  */
}
```

String-Conversion Functions <stdlib.h>

Prototype	Function Description
double atof (const char* str);	Convert string to double
int atoi (const char * str);	Convert string to integer

```c
/* atoi example */
#include <stdio.h>      /* printf, fgets */
#include <stdlib.h>     /* atoi */

int main ()
{
  int i;
  char buffer[256];
  printf ("Enter a number: ");
  fgets (buffer, 256, stdin);
  i = atoi (buffer);
  printf ("The value entered is %d. Its double is %d.\n",i,i*2);
  return 0;
  /*
  Output:
    Enter a number: 5
    The value entered is 5. Its double is 10.
  */
}
```

String-Manipulation Functions <string.h>

Prototype	Function Description
char * strcat (char * destination, const char * source);	Concatenate strings
char * strcpy (char * destination, const char * source);	Copy string
int strcmp (const char * str1, const char * str2);	Compare two strings

```c
/* strcat, strcpy example */
#include <stdio.h>
#include <string.h>

int main ()
{
  char str[80];
  strcpy (str,"these ");
  strcat (str,"strings ");
  strcat (str,"are ");
  strcat (str,"concatenated.");
  puts (str);
  return 0;
  /*
  Output:
    these strings are concatenated.
  */
}
```

```c
#include <string.h>
#include <stdio.h>

void demo(const char* lhs, const char* rhs)
{
    int rc = strcmp(lhs, rhs);
    const char *rel = rc < 0 ? "precedes" : rc > 0 ? "follows" : "equals";
    printf("[%s] %s [%s]\n", lhs, rel, rhs);
}

int main(void)
{
    const char* string = "Apple";
    demo(string, "Banana");
    demo(string, "Abc");
    demo(string, "Apple");
}
/*
Output:
    [Apple] precedes [Banana]
    [Apple] follows [Abc]
    [Apple] equals [Apple]
*/
```

Exercise 8

- □□□
  □□ input □□□□(□□□)□□□□□□□□, □□□□□□□□□□□□□
- Input

```
-619Nri-805vE559z-478S284zs560n
658q-692Z-327HNMJ31Pd-763j-92b
809ZG-307SB459E-82l748XT-120jp
```

- Output

```
-499
-1185
1507
```

Any Question?