Computer Programming 1 Lab

2022-11-24

Mozix Chien

Outline

- Character
- String
- Recursive Function
- Exercise

Character

Character

Declare

```
char c;
char c1 = 'c';
char c2 = '1';
```

- Using `''` means that it is a character, even if it looks like number.

Character

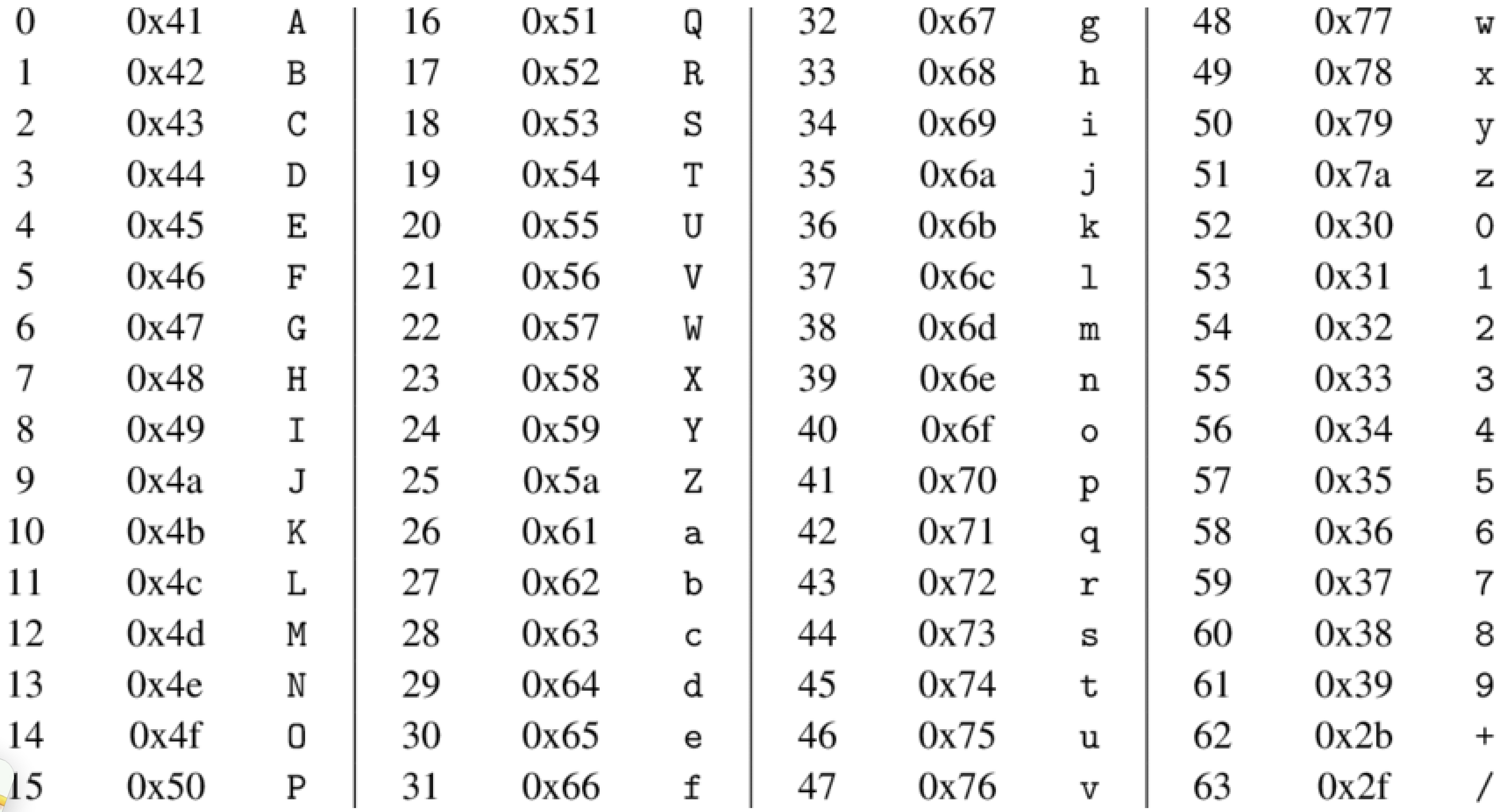
I/O

```cpp
char c;

// Input
cin >> c;
scanf("%c", &c);

// Output
cout << c << endl;
printf("%c", c);
```

- While using **scanf** or **printf**, `%c` will make compiler see anything as character.

Character

ASCII Code



0	0x41	A	16	0x51	Q	32	0x67	g	48	0x77	w
1	0x42	B	17	0x52	R	33	0x68	h	49	0x78	x
2	0x43	C	18	0x53	S	34	0x69	i	50	0x79	y
3	0x44	D	19	0x54	T	35	0x6a	j	51	0x7a	z
4	0x45	E	20	0x55	U	36	0x6b	k	52	0x30	0
5	0x46	F	21	0x56	V	37	0x6c	l	53	0x31	1
6	0x47	G	22	0x57	W	38	0x6d	m	54	0x32	2
7	0x48	H	23	0x58	X	39	0x6e	n	55	0x33	3
8	0x49	I	24	0x59	Y	40	0x6f	o	56	0x34	4
9	0x4a	J	25	0x5a	Z	41	0x70	p	57	0x35	5
10	0x4b	K	26	0x61	a	42	0x71	q	58	0x36	6
11	0x4c	L	27	0x62	b	43	0x72	r	59	0x37	7
12	0x4d	M	28	0x63	c	44	0x73	s	60	0x38	8
13	0x4e	N	29	0x64	d	45	0x74	t	61	0x39	9
14	0x4f	O	30	0x65	e	46	0x75	u	62	0x2b	+
15	0x50	P	31	0x66	f	47	0x76	v	63	0x2f	/

Character

ASCII Code

```c
int c1 = 'a', c2 = 'A', c3 = ' ';
printf("%c, %d\n", c1, c1); // a, 97
printf("%c, %d\n", c2, c2); // A, 65
printf("%c, %d\n", c3, c3); //  , 32
```

- Because of ASCII Code, we can't input chinese in C/C++
- Other Codes like Utf-8, Unicode, etc... are using in other programming languages.

String

String

Definition

> In C/C++, a string is a char array.

```c
char str1[10];
char str2[10] = "I like C";
char str3[10] = {'J', 'o', 'h', 'n'};
char str4[] = "I don't like C";
```

- Using `""` means that it is a string, note that `""` is different from `''`.

String

```c
char str2[4] = "John";
// error: initializer-string for char array is too long
```

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

String

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]);
  }
  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";
 while (str[i++]) {
  putchar (toupper(str[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 ()
{
  char str[256];
  printf ("Enter a number: ");
  gets (str);
  int num = atoi (str);
  printf ("The value entered is %d. Its double is %d.\n",num,num*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
int strlen (const char * str1);	string length

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

int main ()
{
  char str[80];
  strcpy (str,"Hi ");
  strcat (str,"john");
  strcat (str,"cena!");
  puts (str);
  return 0;
  /*
  Output:
    Hi johncena!
  */
}
```

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

void demo(const char* str1, const char* str2)
{
    if (strcmp(str1, str2) == 0)
        puts("equal");
    else
        puts("not equal");
}

int main(void)
{
    const char* string = "Apple";
    demo(string, "Banana");
    demo(string, "Abc");
    demo(string, "Apple");
}
/*
Output:
    not equal
    not equal
    equal
*/
```

Recursive Function

> ~~遞迴只應天上有，凡人應當用迴圈~~

Recursive Function

Fibonacci

```
1 1 2 3 5 8 13 21 ...
```

- $a_1 = 1$
- $a_2 = 1$
- $a_n = a_{n-1} + a_{n-2} \quad (n \geq 3)$

$$fib(n) = \begin{cases} 1 & n = 1 \; or \; n = 2 \\ fib(n-1) + fib(n-2) & n \geq 3 \end{cases}$$

Recursive Function

Fibonacci in Iteration

```
1 1 2 3 5 8 13 21 ...
```

```
fib[n] = {};
fib[1] = 1, fib[2] = 1;

for(int i=3;i<n;i++)
  fib[i] = fib[i-1] + fib[i-2];
```

Recursive Function

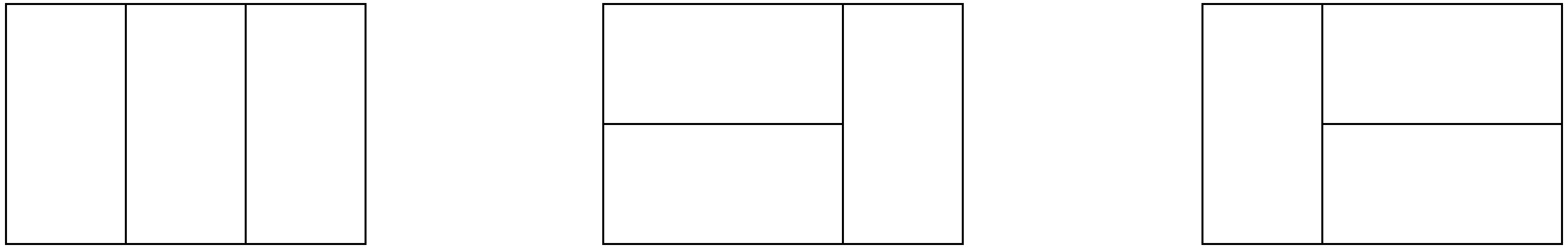
Fibonacci in Recursive

```
int fib(int n){
    if(n == 0 || n==1)
        return 1;
    else return fib(n-1) + fib(n-2);
}
```

Recursive Function

貼磁磚

- 你想知道用1×2的磁磚貼滿2×n的牆壁有幾種方式。



Recursive Function

貼磁磚

- 你想知道用1×2的磁磚貼滿2×n的牆壁有幾種方式。

試著整理一下～

- $n = 1, ans = 1 \implies f(1) = 1$
- $n = 2, ans = 2 \implies f(2) = 2$
- $n = 3, ans = 3 \implies f(3) = 3$
- $n = 4, ans = 5 \implies f(4) = 5$

  ...

Recursive Function

貼磁磚

- 你想知道用1×2的磁磚貼滿2×n的牆壁有幾種方式。
- 這不就是非波那契數列嗎？

```
int ans[1000]={0,1,2};

int solve(int n){
    if(n<2)
        return ans[n];
    return ans[n-1] + ans[n-2];
}
```

Recursive Function

計算冪

- 給定兩個數字 n, k，求 n^k

Recursive Function

計算冪

- 給定兩個數字n, k，求n^k

$$n \times n \times n \times \dots \times n \times n \times n$$

```
int ans = 1;
for(int i=0;i<k;i++) // this iteration will do k times, too slow => O(n)
  ans *= n;
```

Recursive Function

計算冪

- 給定兩個數字n, k，求n^k

$$[(n \times n) \times (n \times n)] \times \ldots \times [(n \times n) \times (n \times n)]$$

- k = 8

$$(n \times n) \times (n \times n) \times (n \times n) \times (n \times n)$$

$$(n^2 \times n^2) \times (n^2 \times n^2)$$

$$(n^4 \times n^4)$$

$$n^8$$

Recursive Function

計算冪

- 給定兩個數字n, k，求n^k

```
int pow(int n, int k){
    if(k == 1)
        return n;
    else if(k%2 == 0)
        return pow((n*n), k/2);
    else
        return (pow((n*n), k/2)*n);
}
```

Recursive Function

其他應用

- GCD (Greatest Common Divisor)
- Complete Search in DFS (Depth First Search)
- Quicksort (Divide and Conquer)

Exercise

Exercise

電影《三個傻瓜》中，拉加為了考試順利，不惜配戴許多作弊用品聖戒，做許多的法事。雖然在藍丘的脅迫循循善誘下，他終於擺脫了滿手的聖戒。但……一切都是暫時的。最近，他迷上了另一種祈禱術，只要他能在1秒內正確地說出刻印在石碑上聖痕(#)的個數，他將無往不利。可惜好友藍丘正在度蜜月，無法找他幫忙，於是他找上了聰明的你幫忙。

- Input:
  第一行有一個正整數 T $(T \leq 1000)$ 接著 T 行，每行各有一個n $(n \leq 20)$，表示石碑的大小 $(n \times n)$ 。接著讀入一張圖，保證輸入只由'#', '.'所組成。
- Output:
  輸出總共有多少聖痕。
- Hint:
  DFS in Recursive Function

Any Questions?