

[11] 配列 (2) -文字と文字列-

テキストp.38-44

1. 文字型変数

文字型変数の宣言	char	char a, code;
文字型の書式制御	%c (1文字)	printf("%c\n",code);
文字列の書式制御	%s	printf("%s\n","program");
文字型変数のサイズ	1バイト	1文字分

☆文字定数は ('') でかこむ. 'A'. 文字列定数場合は (") でかこむ. "sample"

2. 文字コード

・コンピュータで使用する文字には文字コード (アスキーコード) が決まっている. 一般的に, 文字コード32~127に, 英字, 記号, 数字 (半角) が対応している.

3. 文字列の入出力

☆文字型変数は「1文字」のみ. 「文字列」を記憶するには文字型配列変数が必要.

※文字列を入力すると, 文字列の終わりに\0 (nullコード) が付け足されるので, 記憶する文字数+1の大きさ (要素数) の文字型配列を宣言しておく.

文字列の入出力関数**○書式付き入出力**

printf(format [,arg1,...])	書式には, %sを用いる.
scanf(format, arg1,...)	

☆scanf関数では, 「空白 (スペース)」を含んだ文字列を入力できない.

空白を含む文字列を入力したい場合は, gets関数を用いる

○1行単位の入出力

puts(buf);	1行出力. buf 文字型配列変数名
gets(buf);	1行入力. 入力終了 (^z) でNULLを返す.

例) char buf[256];	文字型配列変数bufを宣言.
gets(buf);	bufに文字を入力する.
puts(buf);	bufの内容を出力する.

○1文字入出力

putchar(c);	1文字出力. cは文字型変数.
c=getchar();	1文字入力. 入力終了 (^z) でNULLを返す.

☆puts, gets, putchar, getcharでは, 最初に #include <stdlib.h> が必要.

例) 文字列を利用したプログラム例 (入力した文字列を1文字ずつ表示する)

```
int i=0;
char str[101];           100文字まで入力する場合, 要素数は101.
scanf("%s",str);        キーボードから文字列を入力.
printf("%s \n",str);    入力した文字列を表示する.
while(str[i]!='\0'){    文字列の最後 (\0) までを表示する.
    printf("%c \n",str[i]);  入力した文字列を1文字ずつ表示する.
    i++;
}
```

- 空白を含む文字列を扱う場合はscanfをgetsに変更する.

4. 文字型配列の初期化

- 文字型配列のデータを初期化する (文字列を代入する) 場合, 次のように書く.

static char 配列名 [要素数] =文字列;

例) static char str[100] = "Sample"; 要素数は省略することができる.