Chapter 9 반복문(for문, while문 )

For문

– 특정한문장을일정한횟수만큼반복시킬때사용

For문의 형식과 실행순서

for(초기화; 조건식; 증감 식) {

❶ ❷ ❹

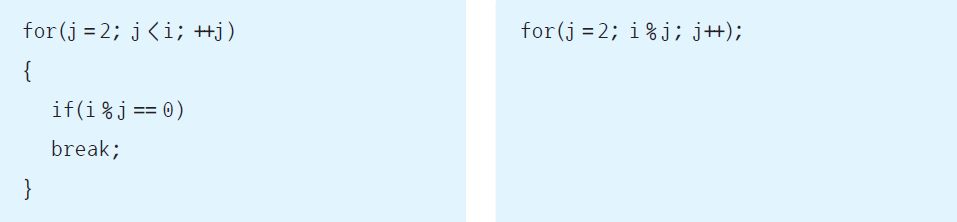
❸ 명령문 1;

명령문 2;

⋮

}

* –  초기화 : 변수의 초기값을 지정, 반복문의 시작을 나타냄
* –  조건식 : 변수값이 조건식에 맞으면 반복을 계속함
* –  명령문들 : 조건식이 참이면 명령문들을 실행함
* –  증감식 : 증감식에 따라 변수값을 증가시키거나 감소시킴
* –  실행순서 : for문은 1 → 2 → 3 → 4 → 2 → 3 → 4 ...순으로 실행됨
* –  종료조건:조건식이거짓이될때종료함



• 왼쪽코드

– 이전 [예제]에서 두 번째 for문을 i가 소수인지 아닌지 판단하려고 작성한 것

• 오른쪽코드

– 오른쪽의 for문은 실행문이 없는 특이한 경우

– 아무일도하지않지만종료조건이i%j이므로나머지가0일경우에 종료됨

– 소수인 경우는 i값과 j값이 동일함

1.

int main() {

for (int i = 0; i < 5; i++) {

printf("\*\*\*\*\*\*\*\*\*\*\n" );

}

return 0;

}

2.

#include <stdio.h>

int main() {

int sum =0;

for (int i = 1; i <=10; i++) {

sum+=i;

}

printf("1부터 10까지의 합은 %d다.\n",sum );

return 0;

}

3.

#include <stdio.h>

int main() {

int number;

printf("반복 횟수를 입력하세요 : " );

scanf("%d",&number );

for (int i = 1; i <= number; i++) {

printf("%d번 반복입니다!!\n",i );

}

return 0;

}

4.

#include <stdio.h>

int main() {

for (int i = 0; i < 5; i++) {

printf("\*\*\*\*\*\*\*\*\*\*\n" );

}

printf("\n\n" );

for (int i = 1; i < 10; i+=2) {

printf("\*\*\*\*\*\*\*\*\*\*\n" );

}

printf("\n\n" );

for (int i = 10; i > 0; i-=2) {

printf("\*\*\*\*\*\*\*\*\*\*\n" );

}

printf("\n\n" );

return 0;

}

5.

#include <stdio.h>

int main() {

for (int i = 65; i <=90; i++) {

printf("%c ",i );

}

printf("\n" );

return 0;

}

6.

#include <stdio.h>

int main() {

for (int i = 0, j=0; i < 3; i++, j++) {

printf("i = %d, j = %d\n",i,j );

}

return 0;

}

7.

#include <stdio.h>

int main() {

int sum;

printf("다중 for문 예제입니다.\n" );

for (int i = 1; i <= 3; i++) {

for (int j = 1; j <= 3; j++) {

sum=i+j;

printf("%d + %d = %d\n",i,j,sum );

}

}

return 0;

}

8.

#include <stdio.h>

int main() {

int sum;

for (int i = 1; i < 6; i++) {

for (int j = 0; j < i; j++) {

printf("\*" );

}

printf("\n" );

}

return 0;

}

9.

#include <stdio.h>

int main() {

int input, index;

printf("임의의 정수를 입력하세요 : " );

scanf("%d",&input );

for (int i = 1; i <= input; i++) {

index=0;

for (int j = 1; j <= i; j++) {

if (i%j==0) {

index++;

}

}

if (index==2) {

printf("%d\n",i);

}

}

printf("\n\n" );

return 0;

}

10.

#include <stdio.h>

int main() {

int count=0;

for (int i = 2; i <= 100; i++) {

for (int j = 2; j <= i; j++) {

if (i%j==0) {

if (j==i) {

count++;

printf("%d%s",i,count%5?" " : "\n" );

}

break;

}

}

}

printf("\n\n" );

return 0;

}

11.

#include <stdio.h>

int main() {

int j;

int count=0;

for (int i = 2; i <= 100; i++) {

for (j = 2; i%j; j++);

if (j==i) {

count++;

printf("%d%s",i,count%5?" " : "\n" );

}

}

return 0;

}

12.

#include <stdio.h>

int main() {

int sum= 0, number=1;

printf("1에서 10까지 합을 구합니다.\n" );

while (number<=10) {

sum+=number;

number++;

}

printf("합은 %d입니다.\n",sum );

return 0;

}

13.

#include <stdio.h>

int main() {

int number=65;

while (number<=90) {

printf("%c ",number );

number++;

}

return 0;

}

14.

#include <stdio.h>

int main() {

int sum= 0, number=0;

while (sum<300) {

number++;

sum+=number;

}

printf("합이 300이 넘는 a의 최솟값은 %d다.\n",number );

printf("sum = %d\n", sum);

return 0;

}

Quiz 9\_1

#include <stdio.h>

int main() {

for (int i = 5; i > 0; i--) {

for (int j = 0; j < i; j++) {

printf("\*" );

}

printf("\n\n");

}

for (int i = 3; i >0 ; i--) {

for (int j =0; j < 2\*i-1; j++) {

printf("\*" );

}

printf("\n\n");

}

/\*

for (int i = 5; i >0 ; i-=2) {

for (int j =0; j < i; j++) {

printf("\*" );

}

printf("\n\n");

}\*/

return 0;

}

Quiz 9\_2

#include <stdio.h>

int main() {

for (int i = 1; i < 10; i++) {

printf(" %d 단 ",i );

}

printf("\n" );

for (int i = 1; i < 10; i++) {

for (int j = 1; j < 10; j++) {

if (j\*i>=10) {

printf("%dx%d=%d ",j,i,j\*i );

} else{

printf("%dx%d= %d ",j,i,j\*i );

}

//printf("%dx%d=%2d ",j,i,j\*i );

}

printf("\n" );

}

return 0;

}

Quiz 9\_3

#include <stdio.h>

int main() {

int input;

printf("1~9사이의 정수를 입력하시오 : " );

scanf("%d",&input );

for (int i = 0; i <= input; i++) {

for (int j = 0; j < i; j++) {

printf("%d",j+1 );

}

printf("\n\n" );

}

return 0;

}

Chapter 10 (do~while문 break, continue, goto)

goto문  
– 실행 순서를 사용자가 지정한 라벨로 이동시킴  
– 명령문의 실행 순서를 프로그래머가 임의로 변경하고자 할 때 사용함 – goto문은 실행 순서를 지정된 라벨로 옮겨 프로그램을 계속 수행함

• 기본형식  
– 라벨(Label)은 프로그램의 다른 명령문 앞에 선언함

1.

#include <stdio.h>

int main() {

int sum= 0, number=0;

printf("1에서 10까지 합을 구합니다.\n" );

do {

number++;

sum+=number;

} while(number<10);

printf("합은 %d입니다.\n",sum );

return 0;

}

2.

#include <stdio.h>

int main() {

int sum= 0, number=0;

do {

number++;

sum+=number;

} while(sum<1000);

printf("합이 1000보다 작은 최대합은 = %d\n",sum-number );

printf("마지막으로 더해지는 자연수는 %d다.\n",number-1 );

return 0;

}

3.

#include <stdio.h>

int main() {

int sum= 0, number=0;

do {

number+=7;

sum+=number;

printf("%d %d\n",sum,number );

} while(sum<100);

printf("누적합 : %d 배수 : %d\n",sum,number);

return 0;

}

4.

#include <stdio.h>

int main() {

int sum=0,number=0;

while (1) {

sum+=number;

number++;

if (number>10) {

goto ex1;

}

}

ex1:

printf("1부터 10까지 합을 구합니다.\n" );

printf("합은 %d입니다.\n",sum );

return 0;

}

5.

#include <stdio.h>

int main() {

for (int i = 1; i <=100; i++) {

for (int j = 1; j <= 9; j++) {

printf("%dx%d=%2d\n",i,j,i\*j );

if (i==9&&j==9) {

goto end;

}

}

}

end:

printf("goto문을 빠져 나왔습니다.\n" );

return 0;

}

6.

#include <stdio.h>

int main() {

for (int i = 1; i <= 100; i++) {

printf("%d ",i );

if (i>=5) {

break;

}

}

printf("\nfor문을 빠져나왔습니다.\n" );

return 0;

}

7.

#include <stdio.h>

int main() {

int sum=0;

for (int i = 1; i <= 10; i++) {

if (i%2==0) {

continue;

}

sum+=i;

}

printf("1부터 10까지 홀수의 합은 %d다.\n",sum );

return 0;

}

8.

#include <stdio.h>

int main() {

for (int i = 1; i <= 10; i++) {

if (i%2==0&&i%3==0) {

continue;

}

printf("%d ",i );

}

printf("\n" );

return 0;

}

Quiz 10\_1

#include <stdio.h>

int main() {

int number,count=0;

printf("1~100사이의 정수를 입력하시오 : " );

scanf("%d",&number );

for (int i = number; i >0; i--) {

printf("%3d",i );

count++;

if (count==10) {

printf("\n" );

}

}

return 0;

}

Quiz10\_2

#include <stdio.h>

int main() {

int number,count=0;

printf("1~9사이의 정수를 입력하시오 : " );

scanf("%d",&number );

for (int i = 0; i <number; i++) {

for (int j = number; j > i; j--) {

printf("%d",j );

}

printf("\n" );

}

return 0;

}

Quiz 10\_3

#include <stdio.h>

int main() {

for (int i = 0; i <7; i++) {

for (int j = 1; j <=i+1; j++) {

printf("%3d",(7\*i)+j );

}

printf("\n\n" );

}

return 0;

}

Chapter 11 온라인 시험

1.

#include <stdio.h>

int main() {

int userInput;

printf("1~100사이의 정수를 입력하시오 : " );

scanf("%d",&userInput );

printf("\n" );

int lineCount=0;

for (int i = userInput; i >0 ; i--) {

if (i%2==0) {

printf("%3d",i );

lineCount++;

if (lineCount==10) {

printf("\n" );

}

}

}

return 0;

}

2.

#include <stdio.h>

int main() {

int i;

for (int i = 1; i < 10; i++) {

printf(" %d 단 ",i );

if(i%3==0){

printf("\n" );

for (int j = 1; j <10; j++) {

for (int k = i-2; k < i+1; k++) {

printf("%dx%d=%2d ",k,j,k\*j );

}

printf("\n" );

}

printf("\n" );

}

}

return 0;

}

3.

#include <stdio.h>

int main() {

int userInput;

int sum=0;

printf("1~100사이의 정수를 입력하시오 : " );

scanf("%d",&userInput );

printf("\n" );

for (int i = 1; i <= userInput; i++) {

if (i%2==0) {

if (i%3==0) {

sum+=i;

}

}

}

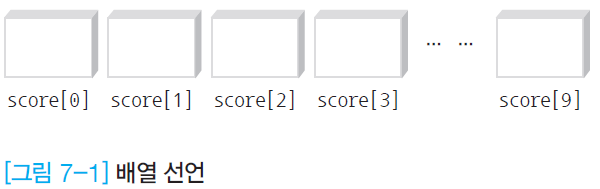
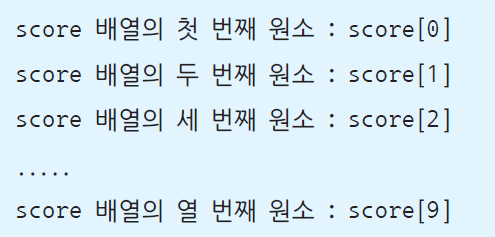
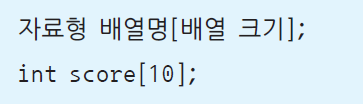
printf("1부터 입력하신 수까지의 짝수이면서 3의 배수인 수들의 합은 %d입니다.\n",sum );

return 0;

}

Chapter 12 (1차원,2차원,3차원 배열)

배열선언

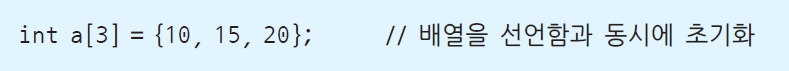
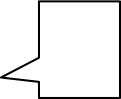
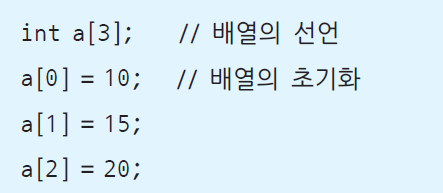


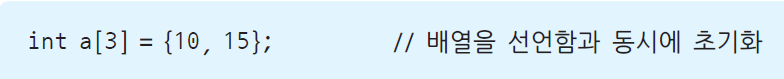
– int형 변수 1개의 메모리 크기가 4바이트 -> score[10]의 메모리 크기는 40바이트.

– 배열의각요소는배열명에원소번호를붙여서명명. – 배열의 첫 번째 원소는 0번 부터 시작 ex) score[0]

배열을먼저선언하고 배열의각 원소값을 할당하는방법

• 배열을 선언하면서 초기화하는 방법.



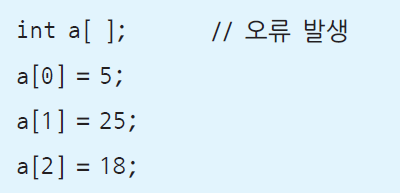
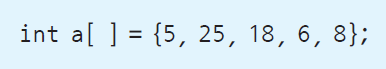
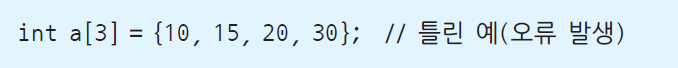


a[0]=10 a[1]=15 a[2]= 0

– 배열을 선언함과 동시에 초기화할 때 배열의 원소보다 초기값을 적게 할당하면, 초기값이 할당되지 않는 배열의 원소에는 0이 저장됨.

* 배열의원소보다초기값을많이할당한예:오류발생
* 원소의 개수를 표시하지 않았지만, 그 개수만큼 배열의 원소가 자동 생성되는 예 - 배열의 원소를 표시하지 않고 초기값을 할당하는 것이 오류 발생을 줄이는 방법이기도 함.
* 배열선언과초기값할당을따로하는예:오류발생

– 원소의 개수를 반드시 기록해야 함.

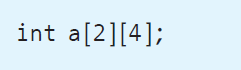
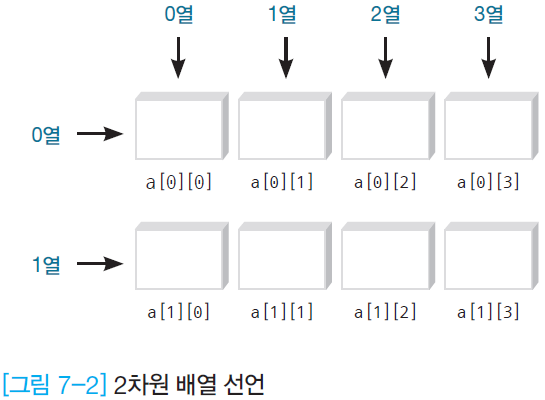


연속적인 저장 공간을 선언하는 배열에서 1차원뿐만 아니라 2차원 이상도 가능.

• 2차원배열의형태:행과열로 표현된 행렬

행열

– 대괄호([ ]) 2개를 이용해서 정의함.  
– a[2][4] : 행이 2개이고 열이 4개인 2차원 행렬

page3image31032064page3image31045696

2차원 배열의 선언

– 중괄호({ })로 배열의 원소를 행 단위로 묶어 초기화하거나, 전체 원소를 묶어 초기화

– 2차원배열을선언할때,행의개수는생략(O).열의개수는생략(X)

• 올바른 예: int a [ ][3] = {10, 20, 30, 40, 50, 60};

• 틀린 예 1: int a [2][ ] = {10, 20, 30, 40, 50, 60};

• 틀린 예 2: int a [ ][ ] = {10, 20, 30, 40, 50, 60};

• 틀린 예 3: int a [ ][3];

• 올바른 예: int a[ ][3] = {10, 20, 30, 40};

2차원 배열의 초기화

– 행의 순서에 따라 2차원 배열이 순서대로 초기화 int num[2][3] = {1, 2, 3, 4, 5, 6};

int num[2][3] = {{1, 2},{3, 4, 5}};

int num[2][3] = {1, 2, 3, 4, 5}; int num[ ][3] = {1, 2, 3, 4, 5};

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| 4 | 5 | 6 |

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 0 |
| 3 | 4 | 5 |

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| 4 | 5 | 0 |

1.

#include <stdio.h>

int main() {

int a[]={5,10,15};

for (int i = 0; i < 3; i++) {

printf("a[%d] = %d\n",i,a[i] );

}

return 0;

}

2.

#include <stdio.h>

int main() {

int A[5]={1,2,3,4,5};

int sum =0;

double average;

for (int i = 0; i < 5; i++) {

sum+=A[i];

}

average=(double)sum/5;

printf("배열 A 원소의 합은 %d다.\n",sum );

printf("배열 A 원소의 평균은 %4.2lf이다.\n", average);

return 0;

}

3.

#include <stdio.h>

int main() {

int a[2][3]={{15,20,30},{35,40,45}};

for (int i = 0; i < 2; i++) {

for (int j = 0; j < 3; j++) {

printf("a[%d][%d] = %d\n",i,j,a[i][j] );

}

}

return 0;

}

4.

#include <stdio.h>

int main() {

int a[2][2]={5,20,12,7};

int sum=0;

for (int i = 0; i < 2; i++) {

for (int j = 0; j < 2; j++) {

sum+=a[i][j];

}

}

printf("2차원 배열 a의 합은 %d다.\n",sum );

return 0;

}

5.

#include <stdio.h>

int main() {

int a[][3]={{90,80,90},{70,100,100},{80,90,90}};

int sum=0;

double avg[3];

for (int i = 0; i < 3; i++) {

for (int j = 0; j < 3; j++) {

sum+=a[i][j];

}

avg[i]=(double)sum/3;

printf("%d번 학생의 평균은 %.2lf이다.\n",i+1,avg[i]);

sum=0;

}

return 0;

}

6.

#include <stdio.h>

int main() {

int a[][3]={{90,80,90},{70,100,100},{80,90,90}};

int sum=0;

double avg[3];

for (int j = 0; j < 3; j++) {

for (int i = 0; i < 3; i++) {

sum+=a[i][j];

}

avg[j]=(double)sum/3;

if (j==0) {

printf("국어 평균은 %.2lf이다.\n",avg[j]);

} else if (j==1) {

printf("영어 평균은 %.2lf이다.\n",avg[j]);

} else if (j==2) {

printf("수학 평균은 %.2lf이다.\n",avg[j]);

}

sum=0;

}

return 0;

}

7.

#include <stdio.h>

int main() {

int x[2][2][5]={0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19};

for (int i = 0; i < 2; i++) {

for (int j = 0; j < 2; j++) {

for (int k = 0; k < 5; k++) {

printf("x[%d][%d][%d] = %d\n",i,j,k,x[i][j][k] );

}

}

}

return 0;

}

8.

#include <stdio.h>

int main() {

int mult[]={4,2,5,1,3};

int result=1;

for (int i = 0; i < 5; i++) {

result\*=mult[i];

}

printf("제시된 모든 원소의 곱은 %d\n",result );

return 0;

}

Quiz 12\_1

#include <stdio.h>

int main() {

int nums[5]={2,19,1,27,16};

int target=nums[0];

for (int i = 0; i < 5; i++) {

printf("nums[%d] is %2d\n",i,nums[i] );

if (target<nums[i]) {

target=nums[i];

}

}

printf("\n최대값은 %d 이다.",target );

return 0;

}

Quiz 12\_2

#include <stdio.h>

int main() {

int Array[3][4]={{10,20,30,40},{-50,-60,-70,-80},{90,100,50,50}};

int sum=0;

double avg;

for (int i = 0; i < 3; i++) {

for (int j = 0; j < 4; j++) {

sum+=Array[i][j];

}

avg=(double)sum/4;

printf("%d행의 합은 %5d, 평균은 %7.3lf \n",i+1,sum,avg );

//소수점도 출력 칸 하나를 차지한다.

sum=0;

}

printf("\n" );

for (int j = 0; j < 4; j++) {

for (int i = 0; i < 3; i++) {

sum+=Array[i][j];

}

avg=(double)sum/3;

printf("%d열의 합은 %5d, 평균은 %7.3lf \n",j+1,sum,avg );

//소수점도 출력 칸 하나를 차지한다.

sum=0;

}

return 0;

}

Chapter 13(문자 배열, 배열주소)

문자 배열(문자열) : char형 문자를 여러 개 모아놓은 문자 집합  
– 문자:작은따옴표‘’  
– 문자열:큰따옴표“”  
– ‘\0’ (NULL) : 문자열 끝에 항상 문자코드 ‘\0’을 덧붙여야 함 배열의 크기를 실제 문자의 개수+1로 선언해야 하는 이유

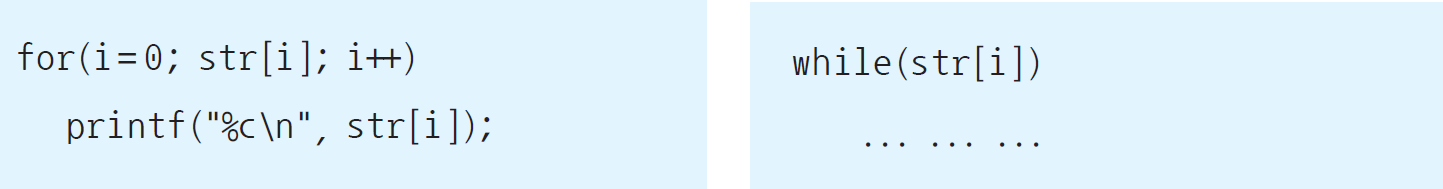
* 올바른 선언: char str[5] = {'A', 'B', 'C', 'D', '\0'}; 잘못된 선언: char str[4] = {'A', 'B', 'C', 'D'};
* 올바른 선언: char str[ ] = {'A', 'B', 'C', 'D', '\0'}; 올바른 선언: char str[ ] = “ABCD”;

• 선언 후 값을 할당 page3image30915968char str[5]; str[1] = 'B';

str[3] = ' D';

str[0] = 'A'; str[2] = 'C'; str[4] = ' \0';

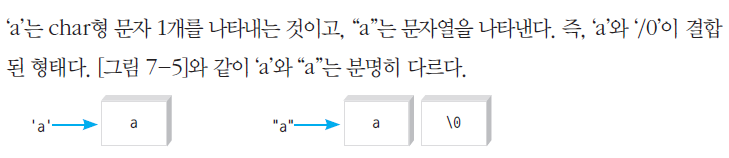
반복 문에서 문자열의 종료 조건



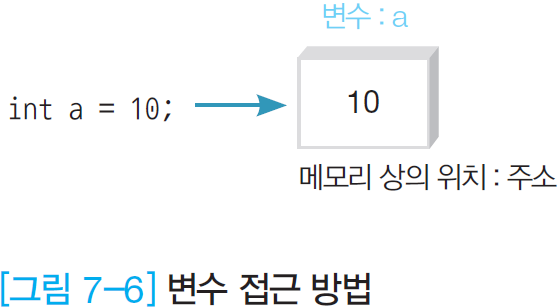
– 문자열을한글자씩출력할경우,반복문의종료조건을위의오른쪽과 같이 while문의 형태로도 많이 표현함

– while(str[i])는 str[i] 값이 ‘\0’이 아니면 계속 반복

문자와 문자열의 차이



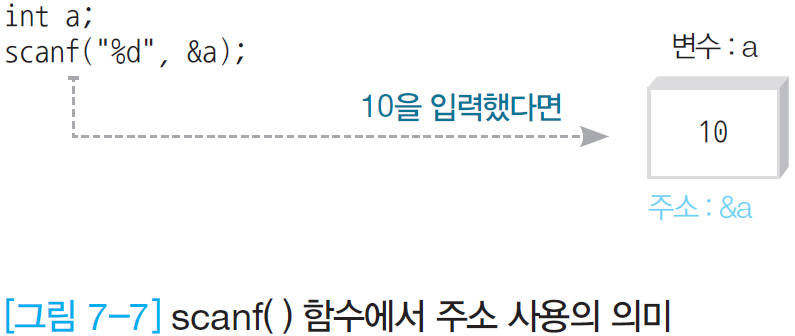
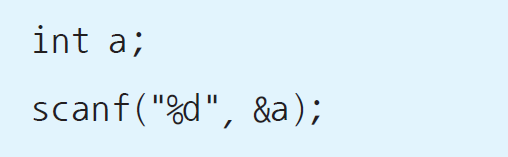
메모리가 할당된 변수에 접근하는 방법



변수의 주소 :변수가 메모리에서 만들어진 위치를 나타냄 유일한 값으로, 변수 앞에 &를 붙여 표시함.

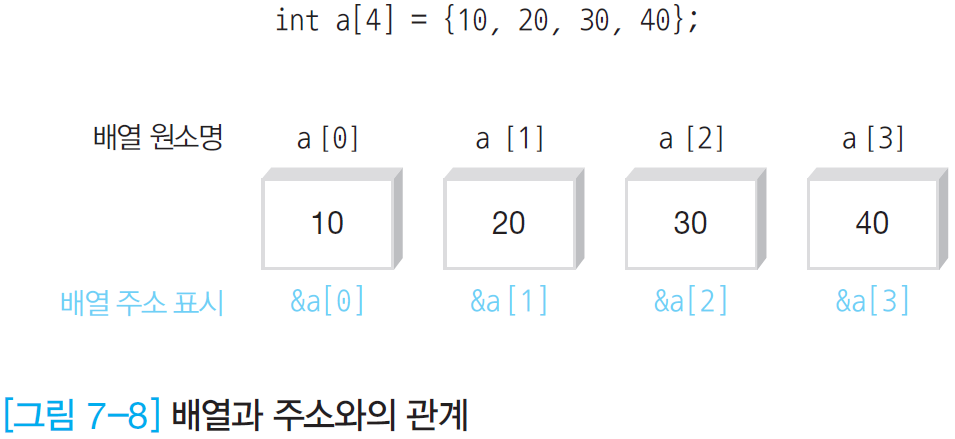
– 변수명으로접근  
– 변수의주소값으로접근:변수앞&를붙임

주소를이용해서변수에접근하는예:scanf()함수

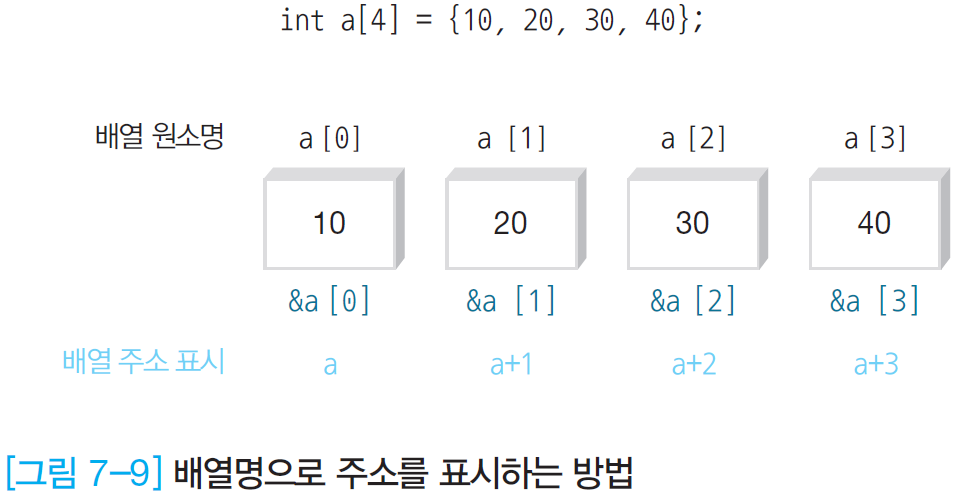


배열과 주소와의 관계

– 배열은동일한자료형을동시에선언하는것  
– 메모리에 배열의 원소 개수만큼의 공간이 동시에 할당됨

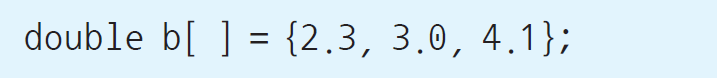
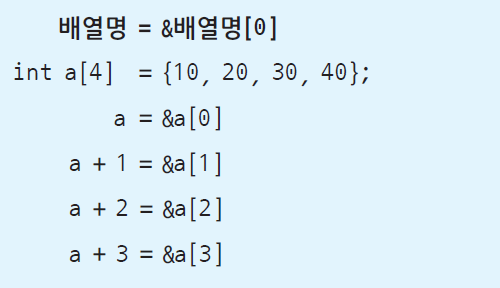


배열 명으로 주소를 표시하는 방법



– 배열명:배열의 시작주소를 나타내는 상수

배열의두번째원소의주소: 배열명+1 – 1은 배열이 선언된 자료형의 크기임



1.

#include <stdio.h>

int main() {

char str[4]={'A','B','C','\0'};

char student[]="John";

printf("문자열을 동시에 출력하는 경우 \n" );

printf("학생의 이름은 %s이다.\n\n",student );

printf("문자열을 한 글자씩 출력하는 경우\n" );

for (int i = 0; str[i]!='\0'; i++) {

printf("%c\n",str[i] );

}

return 0;

}

2.

#include <stdio.h>

int main() {

int a;

printf("정수 값 1개를 입력하시오.\n" );

scanf("%d",&a );

printf("입력받은 정수의 값은 %d이다.\n",a );

printf("입력받은 정수의 주소는 %p이다.\n",&a );

return 0;

}

3.

#include <stdio.h>

int main() {

int a[4]={10,20,30,40};

int aryLength=sizeof(a)/sizeof(int);

printf("배열의 원소값을 출력하면 \n" );

for (int i = 0; i < aryLength; i++) {

printf("a[%d] = %d\n",i,a[i] );

}

printf("배열의 원소의 주소를 출력하면 \n" );

for (int i = 0; i < aryLength; i++) {

printf("&a[%d] = %p\n",i,&a[i] );

}

return 0;

}

4.

#include <stdio.h>

int main() {

int a[4]={10,20,30,40};

double b[ ] = {2.3, 3.0, 4.1, 5.5};

int aryLenInt=sizeof(a)/sizeof(int);

int aryLenDouble = sizeof(b) / sizeof(double);

printf("double형 배열 원소의 주소를 출력하면 \n");

for (int i = 0; i < aryLenDouble; i++) {

printf("b + %d = %p\n",i,b+i );

}

printf("int형 배열의 원소의 주소를 출력하면 \n" );

for (int i = 0; i < aryLenInt; i++) {

printf("a + %d = %p\n",i,a+i );

}

return 0;

}

Quiz 13\_1

#include <stdio.h>

int main() {

int frequency[10]={0};

int data[20];

for (int i = 0; i < 20; i++) {

printf("%2d 번째 정수 : " ,i+1);

scanf("%d",&data[i] );

frequency[data[i]]++; //0~9까지니까 이런식으로 해도 상관없다.

}

printf("\n" );

int max=frequency[0];

for (int i = 0; i < 10; i++) {

if (max<frequency[i]) {

max=frequency[i];

}

}

for (int i = 0; i < 10; i++) {

if (max==frequency[i]) {

printf("정수 :%2d, 최대 빈도수 :%2d회 \n",i,max );

}

}

return 0;

}

Quiz 13\_1\_1

#include <stdio.h>

int main() {

int a[20];

for (int i = 0; i < 20; i++) {

printf("%2d 번째 정수 : " ,i+1);

scanf("%d",a+i );

}

printf("\n" );

int num\_count\_list[10];

int count;

for (int i = 0; i < 10; i++) {

count=0;

for (int j = 0; j < 20; j++) {

if (i==a[j]) {

count++;

}

}

num\_count\_list[i]=count;

}

int target=0;

for (int i = 0; i < 10; i++) {

if (target<num\_count\_list[i]) {

target=num\_count\_list[i];

}

}

for (int i = 0; i < 10; i++) {

if (target==num\_count\_list[i]) {

printf("정수 :%2d, 최대 빈도수 :%2d회 \n",i,target );

}

}

return 0;

}

Quiz 13\_2

#include <stdio.h>

int main() {

char a[4]="";

int i;

char b[4]="1234";

int correct=0;

while (correct==0) {

printf("비밀번호 4자리를 입력하시오! \n" );

scanf("%s",a);

for ( i = 0; i < 4; i++) {

if(a[i]!=b[i]){

break;

}

}

if (i==4) {

correct=1;

} else{

continue;

}

}

printf("비밀번호가 일치합니다!\n" );

return 0;

}

Quiz13\_2\_1

#include <stdio.h>

int main() {

char secret[]="1234";

char input[10];

int number=0;

while (number!=4) {

number=0;

printf("비밀번호 4자리를 입력하시오! \n" );

scanf("%s",input);

for (int i = 0; i < 4; i++) {

if (input[i]==secret[i]) {

number++;

}

}

}

printf("비밀번호가 일치합니다!\n" );

return 0;

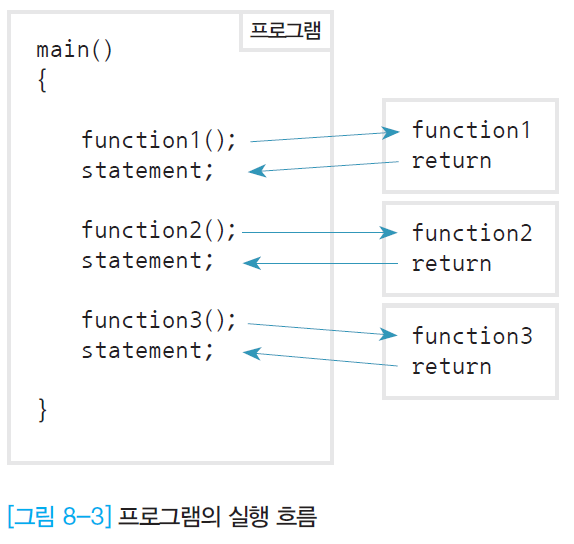
}

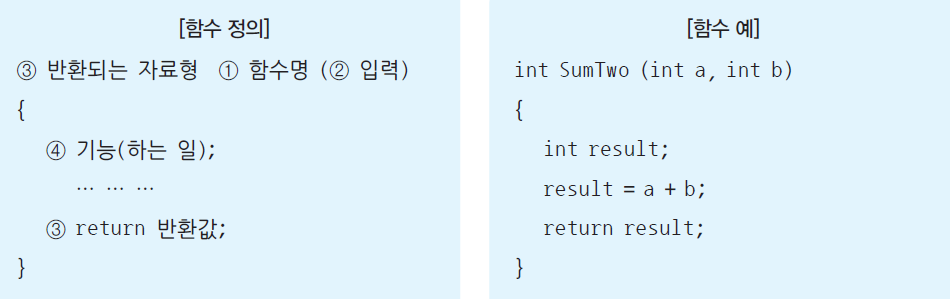
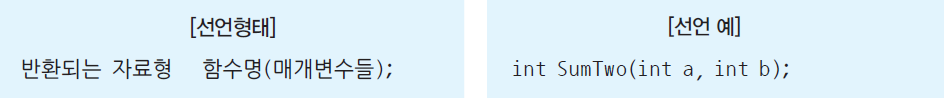
Chapter 14(함수 선언, 호출)

함수:특정한기능을수행하고그수행결과로값을반환하는부프로그램

• 프로그램 실행 흐름  
– 프로그램 실행 중에 함수명과 동일한 부분을 만나면,

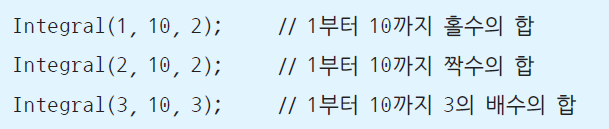
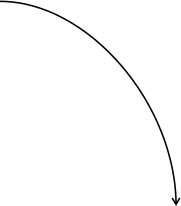
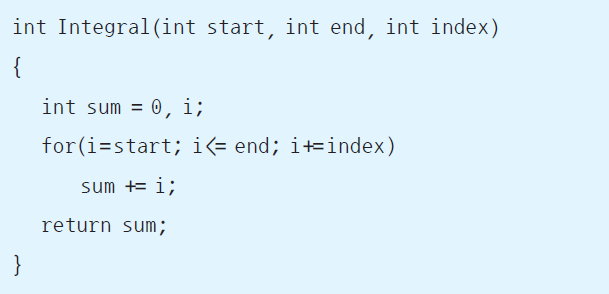
해당함수로 이동 함수 실행 main( ) 함수로 다시 돌아옴.

page5image31062080page5image31059776

* 시스템 라이브러리 함수 :  
  사용자가 바로 이용할 수 있도록, 자주 사용하는 함수를 미리 만들어 놓은 것
* 사용자정의함수:
* 사용자자신이원하는기능을수행 하도록 직접 만든 함수
* 함수의 정의 형태
* 
* – 함수명:식별자작성규칙에맞게작성  
  – 입력:함수에어떤인자가입력되는지나타냄  
  – 반환값:함수가입력을받아기능을수행한후에반환되는결과값 – 기능:함수를통해서사용자가구현하려는내용
* 함수의 선언 : 컴파일러에 함수명과 반환값 그리고 함수의 매개변수를 알려줌
* – 함수 원형(prototype)이라고도 함. – 함수를 정의하기 전에 선언함
* 
* 구체적인 인자 명 대신, 다음과 같이 자료형과 개수만 나타내도 된다. ->>>>>>>>>>>>>>>
* int Sum(int, int)

일반적인 형태로 작성한 함수

• 세번째인자로시작값과종료값사이의간격을정해주는요소(index)추가



1.

#include <stdio.h>

int SumTwo(int x, int y);

int main() {

int x=10,y=5;

int value;

value=SumTwo(x, y);

printf("두 수의 합 : %d\n",value );

return 0;

}

int SumTwo(int x, int y) {

return x+y;

}

2.

#include <stdio.h>

float Average(float x, float y);

int main() {

float x,y;

float result;

printf("실수 2개를 입력하세요 : " );

scanf("%f%f",&x,&y );

result=Average(x, y);

printf("두 실수의 평균 : %f\n",result );

return 0;

}

float Average(float x, float y){

float result=(x+y)/2;

return result;

}

3.

#include <stdio.h>

int Integral(int start, int end);

int main() {

int result=0;

printf("함수를 사용하지 않고 합을 구함\n" );

for (int i = 1; i <= 10; i++) {

result+=i;

}

printf("1부터 10까지 합은 %d다.\n",result);

result=0;

for (int i = 7; i <= 17; i++) {

result+=i;

}

printf("7부터 17까지 합은 %d다.\n",result);

printf("\n함수를 사용하여 합을 구함\n" );

printf("1부터 10까지 합은 %d다.\n",Integral(1,10));

printf("7부터 17까지 합은 %d다.\n",Integral(7,17));

return 0;

}

int Integral(int start, int end) {

int sum=0;

for (int i = start; i <= end; i++) {

sum+=i;

}

return sum;

}

4.

#include <stdio.h>

int SumTwo(int x, int y) {

return x+y;

}

int main() {

int x=10,y=5;

int value;

value=SumTwo(x, y);

printf("두 수의 합 : %d\n",value );

return 0;

}

5.

#include <stdio.h>

void upper\_case(char ch);

int main() {

char ch1;

while ((ch1=getchar())!=EOF) {

upper\_case(ch1);

}

return 0;

}

void upper\_case(char ch) {

char res;

if (ch>='a'&&ch<='z') {

res=ch-32;

} else{

res=ch;

}

putchar(res);

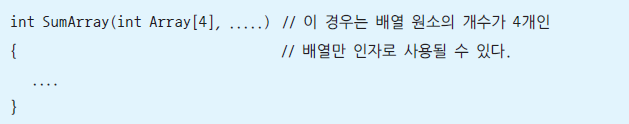
}

Chapter 15 (재귀 함수, 표준함수)

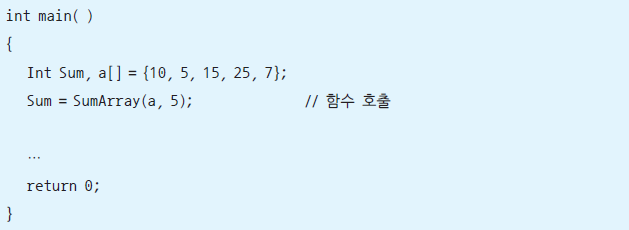
SumArray( ) 함수  
1 배열:배열명은가인자이기때문에식별자작성규칙에따라임의로작성

2 원소개수:일반적으로배열명에는빈대괄호인[]를사용.실제배열의 원소 개수는 두 번째 인자로 표시. 변수 Size가 배열의 원소 개수를 나타냄

3 내용:인자로사용된배열과원소의개수를이용해서필요한기능을작성 4 반환 값: 함수에서 계산된 결과값을 반환

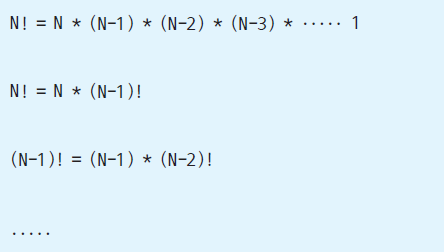


앞의 SumArray( )같이 배열을 인자로 사용하는 함수의 호출



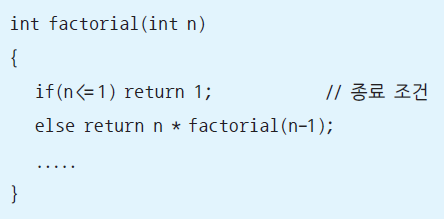
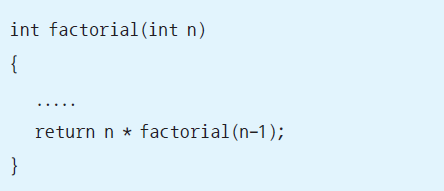
배열 부분에 배열명만 기록

* 재귀호출(Recursion) : 함수에서 그 함수를 다시 호출하는 것
* 재귀함수 : 재귀호출을 구현한 함수



다음 factorial( ) 함수를 보면, 함수에서 자기 자신을 다시 호출

• factorial()함수는n값이1일때나0일때,즉1!이나0!일때 값이 1이므로 다음과 같이 종료 조건을 작성



1,

#include <stdio.h>

int SumArray(int a[], int size);

int main() {

int a[]= {10,5,15,25,7};

int Sum;

Sum=SumArray(a,5);

printf("배열 원소의 합 : %d\n",Sum );

return 0;

}

int SumArray(int a[], int size) {

int result=0;

for (int i = 0; i < size; i++) {

result+=a[i];

}

return result;

}

2.

#include <stdio.h>

int factorial(int n);

int main() {

int fact\_num;

fact\_num=factorial(10);

printf("10 팩토리얼 : %d\n", fact\_num);

return 0;

}

int factorial(int n) {

if (n==1) {

return 1;

}

return n\*factorial(n-1);

}

3.

#include <stdio.h>

int sum(int n);

int main() {

int input, s=0;

printf("양의 정수를 입력하세요 : ");

scanf("%d",&input );

s=sum(input);

printf("재귀함수를 이용한 1부터 %d까지 합 : %d\n", input, s);

return 0;

}

int sum(int n) {

if (n==1) {

return 1;

}

return n+sum(n-1);

}

4.

#include <stdio.h>

int SelectSum(int a[],int size);

int main() {

int a[]={20,10,5,8,28};

int s=0;

s=SelectSum(a, 5);

printf("10보다 큰 배열 원소의 합 : %d\n",s );

return 0;

}

int SelectSum(int a[],int size) {

int result=0;

for (int i = 0; i < size; i++) {

if (a[i]>10) {

result+=a[i];

}

}

return result;

}

stdlib.h or math.h

int abs(int n) int 형 절대값 출력

float fabs(float n) float형 절대값 출력

long labs(long n) long 형 절대값 출력

math.h

double pow(double x, double y) double 형 x^y 출력

double sqrt(double x) double 형 루트(x) 출력

stdlib.h

void srand(unsigned int seed);

sets a random starting point

난수의 초기값 지정

int rand(void); 의사 난수 생성 0~32767

사용방법

stdlib.h

time.h

srand((unsigned)time(NULL));// 난수 초기값 지정

rand()%((max+1)-min)+min // min이상 max이하 난수 출력

ex) 1~9까지 난수 생성

stdlib.h

time.h

srand((unsigned)time(NULL));// 난수 초기값 지정

rand()%9+1 //1이상 9이하 난수 출력