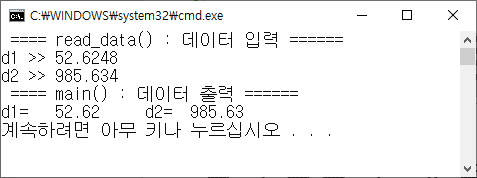
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2020\_1\_C\_8주 | 학번 : | 20175105 | 이름 : | 곽영주 |

* **강의 내용**

1. 간접 참조(\* 연산자 사용)

|  |
| --- |
| [프로그램 소스]  #define \_CRT\_SECURE\_NO\_WARNINGS  #include <stdio.h>  int main() {  int data = 50;  char ch = 'A';  int \*ptrint = &data;  char\* ptrchar = &ch;  printf("간접 참조 출력: %d %c\n", \*ptrint, \*ptrchar);  \*ptrint = 200;  \*ptrchar = 'B';  printf("직접 참조 출력: %d %c\n", data, ch);  printf("주소 값: data ->%p, ptrint -> %p\n", &data, ptrint);  printf("주소 값: ch ->%p, ptrchar -> %p\n", &ch, ptrchar);  printf("메모리크기: ch -> %d, ptrchar -> %d\n", sizeof(ch), sizeof(ptrchar));  return 0;  } |
| [실행 결과] |

1. 매개변수로 포인터 활용



|  |
| --- |
| [프로그램 소스]  #include <stdio.h>  void read\_data(double\*, double\*);  int main() {  double d1, d2;  read\_data(&d1, &d2);  printf(" ==== main() : 데이터 출력 ======\n");  printf("d1=%8.2f\td2=%8.2f\n", d1, d2);  return 0;  }  void read\_data(double \*d1, double \*d2) {  printf(" ==== read\_data() : 데이터 입력 ======\n");  printf("d1 >> ");  scanf\_s("%lf", d1);  printf("d2 >> ");  scanf\_s("%lf", d2);  } |

1. 참조 연산자의 다양한 연산

|  |
| --- |
| [프로그램 소스]  #include <stdio.h>  int main() {  int a[4] = { 1,3,6,8 };  int \*pa = a;  int i;  for (i = 0; i < 4; i++) {  printf("\*(pa + %d) : %d\n", i, \*(pa + i));  }  printf("%d %d %d %d\n", pa[0], pa[1], pa[2], pa[3]);  printf("++\*pa = %d\n", ++\*pa);  printf("\*(++pa) = %d\n", \*++pa);  printf("\*pa++ = %d\n", \*pa++);  printf("--\*pa++ = %d\n", --\*pa++);  printf("(\*pa)-- = %d\n", (\*pa)--);  printf("%d %d %d %d\n", a[0], a[1], a[2], a[3]);  return 0;  } |
| [실행 결과] |

1. 포인터 배열 이용

|  |
| --- |
| [프로그램 소스]  #define \_CRT\_SECURE\_NO\_WARNINGS  #include <stdio.h>  int main() {  int\* pary[3] = { NULL };  int i, a = 10, b = 20, c = 30;  pary[0] = &a;  pary[1] = &b;  pary[2] = &c;  for (i = 0; i < 3; i++)  printf("\*pAry[%d] = %d\n", i, \*pary[i]);  for (i = 0; i < 3; i++) {  printf("%d >> ", i);  scanf("%d", pary[i]);  printf("%d, %d, %d\n", a, b, c);  }  return 0;  } |
| [실행 결과] |

1. 일차원 배열과 포인터

|  |
| --- |
| [프로그램 소스]  #define \_CRT\_SECURE\_NO\_WARNINGS  #include <stdio.h>  int main() {  int ary[] = { 8,2,8,1,3 };  int\* ptr = ary, i;  printf("sizeof(ary) = %d, sizeof(ptr) = %d\n", sizeof(ary), sizeof(ptr));  printf("=== 배열 포인터 : 연산식과 \*를 사용한 배열 원소 참조 ===\n");  for (i = 0; i < 4; i++) {  printf("\*(ptr + %d) = %d\n", i, \*(ptr + i));  }  printf("=== 배열 포인터 : []배열 원소 참조 ==\n");  for (i = 0; i < 4; i++) {  printf("ptr[%d] = %d\n", i, ptr[i]);  }  return 0;  } |
| [실행 결과] |

* **내용 점검**

1. 다음 설명에 대한 변수선언 문장을 작성하시오.
2. short 형 포인터 ps short \*ps;
3. float 형 포인터 pf float \*pf;
4. int 형 포인터 addn int \*addn;
5. double 형 이중 포인터 dpd double \*\*dpd;
6. 모든 자료형에 대한 포인터 p void \*p;
7. int 형 포인터 3개인 배열 pa int \*pa[3];
8. 다음 문장에서 문법 오류 및 실행 오류를 찾아 수정하시오.
9. int a; int \*p = &a;
10. double d; double \*p = &d;
11. int i; int \*pi = &i; int \*\*dpi = &pi;
12. int a, b, c; int \*pf[3] = {&a, &b, &c};
13. char ch[5]; char \*pc = ch;

1. 다음 일차원 배열과 포인터 선언 문장에 대하여 다음 각각의 연산식 결과를 기술하시오.

int score[] = {3, 5, 6, 9, 2, 5, 7, 8, 9, 1};

int \*p = score;

int \*q = p + 4;

1. \*p = 3
2. \*p+1 = 4
3. \*(p+1) = 5
4. ++\*p = 4
5. \*++p = 5
6. \*q = 2
7. q[2] = 7
8. --\*q = 1
9. (\*(q+1))++ = 5
10. \*(q+4) = 9

* **프로그래밍 과제**

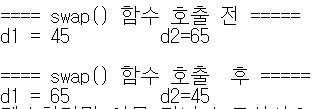
1. 제시된 문제를 해결하는 프로그램을 작성하시오.

* 정수형 변수에 사용자가 입력한 정수를 저장하고 포인터를 이용하여 출력하시오.
* 문자형 변수에 사용자가 입력한 문자를 저장하고 포인터를 이용하여 출력하시오.
* 정수 배열에 사용자가 입력한 정수 다섯 개를 저장하고 포인터를 이용하여 출력하시오.
* 문자 배열에 사용자가 입력한 문자 다섯 개를 저장하고 포인터를 이용하여 출력하시오.

|  |
| --- |
| **[프로그램 소스]**  #include <stdio.h>  int main() {  int i, n, narr[5];  char ch, charr[5];  int \*pn = &n, \*pnarr = &narr;  char \*pch = &ch, \*pcharr = &charr;  printf("정수 입력: ");  scanf\_s("%d", &n);  printf("문자 입력: ");  scanf\_s(" %c", &ch, 1);  printf("입력한 정수 포인터로 출력: %d\n", \*pn);  printf("입력한 문자 포인터로 출력: %c\n", \*pch);  printf("\n=== 배열 ===\n");  for (i = 0; i < 5; i++) {  printf("정수 입력[%d]: ", i);  scanf\_s(" %d", &narr[i]);  }  puts("");  for (i = 0; i < 5; i++) {  printf("문자 입력[%d]: ", i);  scanf\_s(" %c", &charr[i], 1);  }  for (i = 0; i < 5; i++) {  printf("입력한 정수 포인터로 출력[%d]: %d\n", i, \*(pnarr+i));  }  puts("");  for (i = 0; i < 5; i++) {  printf("입력한 문자 포인터로 출력[%d]: %c\n", i, \*(pcharr+i));  }  return 0;  } |
| **[실행 결과]** |

1. 두 개의 데이터를 교환하는 함수를 작성하시오

void swap(int \*a, int \*b) //교환 할 데이터를 매개변수 a, b로 받는다



|  |
| --- |
| **[프로그램 소스]**  #include <stdio.h>  void swap(int \*a, int \*b);  int main() {  int d1 = 45, d2 = 65;    printf("==== swap() 함수 호출 전 =====\n");  printf("d1 = %d \t d2 = %d\n", d1, d2);  swap(&d1, &d2);  printf("==== swap() 함수 호출 후 =====\n");  printf("d1 = %d \t d2 = %d\n", d1, d2);  return 0;  }  void swap(int \*a, int \*b) {  int data = \*a;  \*a = \*b;  \*b = data;  } |
| **[실행 결과]** |

1. 다음을 참고로 배열에서 모든 원소의 값을 모두 n씩 증가시키는 프로그램을 작성하시오.

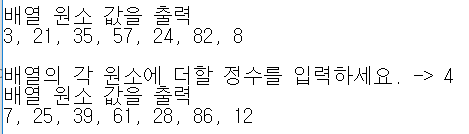
* 배열 int data[] = {3, 21, 35, 57, 24, 82, 8};, n은 표준입력으로 처리
* 함수 increment()는 배열 p에서 배열크기 size만큼 모든 원소의 값을 모두 n씩 증가

void increment(int \*p, int size, int n)

* 함수 print()는 배열 원소 출력

void print(int \*p, int size)

* 함수에서 배열 원소를 참조할 때 \*와 연산식을 사용할 것.



|  |
| --- |
| **[프로그램 소스]**  #include <stdio.h>  void increment(int \*, int, int);  void print(int \*, int);  int main() {  int data[] = { 3,21,35,57,24,82,8 };  int size = sizeof(data) / sizeof(data[0]);  int \*p = data;  int n; // 원소 값 증가량  print(p, size); // n씩 증가 하기 전 출력  puts("");  printf("배열의 각 원소에 더할 정수를 입력하세요. -> ");  scanf\_s("%d", &n);  increment(p, size, n); // n씩 증가  print(p, size); // n씩 증가한 후 출력  return 0;  }  void increment(int \*p, int size, int n) { // 모든 원소 값 n씩 증가  int i;  for (i = 0; i < size; i++)  \*(p + i) += n;  }  void print(int \*p, int size) { // 배열 원소 출력  int i;  printf("배열 원소 값을 출력\n");  for (i = 0; i < size; i++)  printf("%d, ", \*(p + i));  puts("");  } |
| **[실행 결과]** |