|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **C2\_실습과제\_13주** | **이름 :** | **이갑성** | **학번 :** | **20135151** |

* **강의내용**

1. 조건부 컴파일

#include <stdio.h>

#define DEBUG //주석으로 처리하기 전과 후의 결과를 제시

#define LIMIT 20

int main(void) {

int i;

long prod = 1;

for (i = 1; i <= LIMIT; i++){

prod \*= i;

#ifdef DEBUG **//기호상수 DEBUG가 정의되어 있지 않으면 실행되지 않음**

if (i%5 == 0)

printf("DEBUG : 1부터 %d까지의 곱은 %d입니다.\n", i, prod);

#endif

}

printf("1부터 %d까지의 곱은 %d입니다.\n", LIMIT, prod);

return 0;

}

|  |
| --- |
| * 실행결과 |

1. 인자를 사용하는 macro 활용

#include <stdlib.h>

#define MESSAGE "프로그램언어의 학습은 일반언어의 학습과\

같이 반복학습이 중요하다“

#define PI 3.141592 //PI를 3.14로 대치하는 지시자

#define VOLUME(r) (4 \* PI \* CUBE(r) / 3) //구의 체적을 구하는 매크로

#define SQURE(x) ((x) \* (x))

#define CUBE(x) ( SQURE(x) \* (x) )

#define getrandom(min, max) \

((rand()%(int)(((max) + 1)-(min))) + (min))

int main(void) {

double radius = 2.32;

printf("반지름이%.2lf 인구의 체적은 %.2lf 입니다.\n", radius, VOLUME(radius));

printf("실수%.2lf의 제곱은 %.2lf 입니다.\n", 4.29, SQURE(4.29));

printf("실수%.2lf의 제곱은 %.2lf 입니다.\n", 3.0, CUBE(3.0));

printf("%d과%d 사이의 난수: %d\n", 1, 100, getrandom(1, 100));

puts(MESSAGE);

return 0;

}

|  |
| --- |
| * 실행결과 |

1. 전처리 연산자

#define PRT(a) printf(#a" = %d 일때, ", a)

#define APRT(a) printf(#a" = %3d ", a)

#define AIPRT(a, i) printf(#a"[%d] = %3d\n", i, a##[i])

#define CHPRT(a) printf("%c\n", #@a)

int main()

{

int i, prod = 1;

int facto[6];

CHPRT($);

for (i = 1; i <= 5; i++){

prod \*= i;

facto[i] = prod;

PRT(i);

APRT(facto[i]);

AIPRT(facto, i); }

return 0;

}

|  |
| --- |
| * 실행결과 |

* 프로그램 과제

1. “List.txt” 파일에 정리된 링크드 리스트를 실행한 후 결과를 제시하시오

|  |
| --- |
| <소스>  #include "Node.h" //사용자 정의 헤더 파일  NODE \*insert(char item, NODE \*list); //노드 삽입 함수  NODE \*append(NODE \*list, NODE \*temp); //노드 추가 함수  NODE \*delete(char item, NODE \*list); //노드 제거 함수  void print\_list(NODE \*head); //노드를 순회하는 함수  int main() {  NODE \*temp = NULL; //노드 추가를 위해 할당받은 메모리  NODE \*list = NULL; //첫 번째 노드를 가리키는 주소  char ch;  printf("문자를 입력하세요. 숫자0을 입력하면 종료>> \n");  while (1) {  ch = \_getche(); //문자 입력 후 Enter키 입력하지 않음  if (ch == '0')  break;  temp = createNode(ch);  if (temp == NULL) {  exit(1);  }  list = append(list, temp);  }  // 입력 받은 데이터를 추가하여 생성된 연결리스트의 출력  printf("\n");  print\_list(list);  printf("\n");  printf("문자 c을 insert \n");  list = insert('c', list);  print\_list(list);  printf("\n");  printf("문자 h을 insert \n");  list = insert('h', list);  print\_list(list);  printf("\n");  printf("문자 m을 insert \n");  list = insert('m', list);  // 연결리스트 출력  print\_list(list);  printf("\n");  //리스트에서 문자 'h' 삭제  printf("문자 h 삭제 \n");  list = delete('h', list);  print\_list(list);  printf("\n");  return 0;  } |
| <결과> |

1. .

다음의 내용을 구조체에 배열에 초기화 한 후 input.txt로 출력하시오. 또한 input.txt 파일로 출력된 내용을 읽어 링크드 리스트에 저장한 후 화면에 출력하시오. 단, 링크드 리스트 저장 시 번호순으로 저장되도록 한다.

|  |
| --- |
| 1 강동구 30 40  3 강혜진 28 37  5 김다영 32 77  4 김민지 67 39  2 김소연 77 67 |

**[출력예시]**

|  |
| --- |
| 번호 이름 성적1 성적2  1 강동구 30 40  2 김소연 77 67  3 강혜진 28 37  4 김민지 67 39  5 김다영 32 77 |

#include <stdio.h>

#include <stdlib.h>

struct score {

int num;

char name[10];

int sc1;

int sc2;

};

typedef struct score score;

|  |
| --- |
| <소스>  =========Node.h==========  #include <stdio.h>  #include <stdlib.h>  #include <string.h>  struct score {  int num;  char name[10];  int sc1;  int sc2;  struct score \*next;  };  typedef struct score score;  typedef score\* LINK;  =========Main.c===========  #include "Node.h"  LINK createNode(score \*);  LINK insert(LINK cur, LINK head);  void printData(LINK head);  int main()  {  FILE \*f;  score info[5];  score cpy[5];  int i;  int j = 0;  LINK cur = NULL;  LINK head = NULL;  if(fopen\_s(&f, "input.txt", "w") != 0)  {  printf("파일이 열리지 않았습니다.\n");  exit(1);  }  printf("번호, 이름, 중간고사, 기말고사의 성적을 입력하세요 >> \n");  for(i = 0; i < 5; i++)  {  scanf\_s("%d %s %d %d", &((info+i)->num), ((info + i) ->name), 10, &((info + i)->sc1), &((info + i)->sc2));  fprintf(f, "%d %s %d %d \n", ((info+i)->num), ((info + i) ->name), ((info + i)->sc1), ((info + i)->sc2));  }  fclose(f);  if(fopen\_s(&f, "input.txt", "r") != 0)  {  printf("파일이 열리지 않았습니다. \n");  exit(1);  }  for(i = 0; i < 5; i++)  {  fscanf\_s(f, "%d %s %d %d", &((cpy+i)->num), ((cpy + i) ->name), 10, &((cpy + i)->sc1), &((cpy + i)->sc2));  }  fclose(f);  while(j < i)  {  cur = createNode(cpy+j);  if(cur == NULL)  exit(1);  head = insert(cur, head);  j++;  }  printData(head);  }  =========Function.c============  #include "Node.h"  LINK createNode(score \*p)  {  LINK cur = NULL;  if((cur = (score \*)malloc(sizeof(score))) == NULL)  {  printf("메모리 할당에 실패하였습니다. \n");  exit(1);  }  cur->num = p->num;  strcpy(cur->name, p->name);  cur->sc1 = p->sc1;  cur->sc2 = p->sc2;  return cur;  }  LINK insert(LINK input, LINK head)  {  LINK current = NULL; //현재 위치 노드  LINK front = NULL; //현재 노드보다 앞에 위치 노드  current = front = head;  while((current != NULL) && (current->num < input->num))  {  front = current;  current = current->next;  }  input->next = current;    if(current == head)  {  head = input;  }  else  {  front->next = input;  }  return head;  }  void printData(LINK head)  {  LINK cur = NULL; // 현재를 가르키는 노드  printf("===========데이터에 입력된 성적 출력=========== \n\n");  printf("번호 이름 중간고사 기말고사 \n");  for(cur = head; cur; cur = cur->next)  {  printf\_s("%3d %9s %6d %14d \n", (cur->num), (cur->name), (cur->sc1), (cur->sc2));  }  } |
| <결과> |

1. 2번 프로그램에서 번호를 입력하면 해당 레코드를 삭제하는 기능을 추가한 후 결과를 제시하시오.

|  |
| --- |
| <소스>  void delData(LINK head, int num)  {  LINK temp;  LINK current;  LINK front;    front = current = head;  while(current->next != NULL)  {  if(current->num == num)  {  front->next = current->next;  }  front = current;  current = current->next;  }  } |
| <결과> |