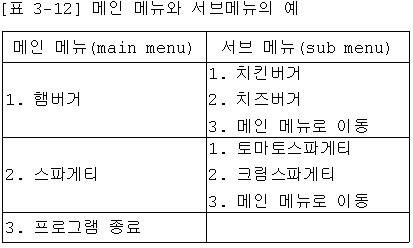
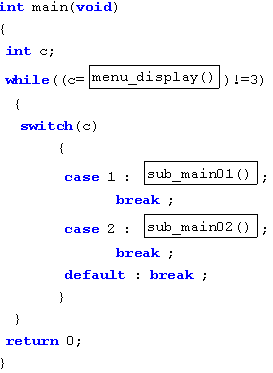
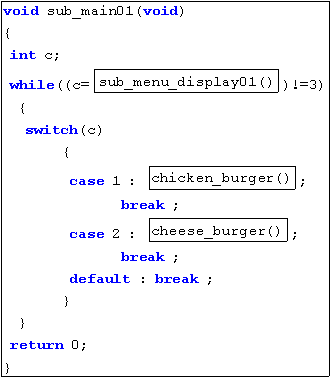
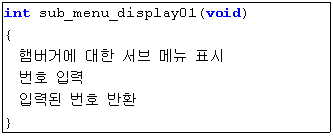
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **C2\_실습과제\_4주** | **이름 :** | **이갑성** | **학번 :** | **20135151** |

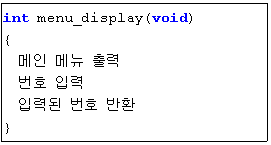
* **강의내용**

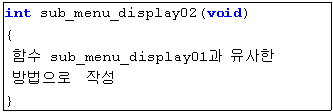
1. 메인 메뉴와 서브 메뉴

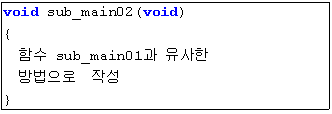
****











* 프로그램 과제 1

1. 메인메뉴와 서브메뉴가 있는 프로그램의 부분코드이다. 함수원형과 생략된 함수를 추가하여 완성하시오

#include <stdio.h>

#include <conio.h>

#include <stdlib.h>

int main(void){

int c;

while ((c = menu\_display()) != 3){

switch (c){

case 1: sub\_main01();

break;

case 2: sub\_main02();

break;

default:

break;

}

}

return 0;

}

int menu\_display(void){

int select;

system("cls");

printf("간식 만들기\n\n");

printf("1. 햄버거 \n");

printf("2. 스파게티\n");

printf("3. 프로그램 종료\n\n");

printf("메뉴번호 입력>");

select = \_getch() - 48;

return select;

}

void sub\_main01(void){

int c;

while ((c = sub\_menu\_display01()) != 3){

switch (c){

case 1: chicken\_burger();

break;

case 2: cheese\_burger();

break;

default:

break;

}}

}

int sub\_menu\_display01(void){

int select;

system("cls");

printf("햄버거 만들기\n\n");

printf("1. 치킨버거\n");

printf("2. 치즈버거\n");

printf("3. 메인 메뉴로 이동\n\n");

printf("메뉴번호 입력>");

select = \_getch() - 48;

return select;

}

void chicken\_burger(void){

system("cls");

printf("치킨버거 만드는 방법\n");

press\_any\_key();

}

void press\_any\_key(void){

printf("\n\n");

printf("아무키나 누르면 이전 메뉴로...");

\_getch();

}

|  |
| --- |
| * 추가된 프로그램 소스   #include <stdio.h>  #include <conio.h>  #include <stdlib.h>  int menu\_display();  void sub\_main01();  int sub\_menu\_display01();  void chicken\_burger();  void press\_any\_key();  void cheese\_burger();  void sub\_main02();  int sub\_menu\_display02();  void tomato\_spa();  void cream\_spa();  int main(void) {  int c;  while ((c = menu\_display()) != 3) {  switch (c) {  case 1: sub\_main01();  break;  case 2: sub\_main02();  break;  default:  break;  }  }  return 0;  }  int menu\_display(void) {  int select; //번호 선택  system("cls");  printf("간식 만들기\n\n");  printf("1. 햄버거 \n");  printf("2. 스파게티\n");  printf("3. 프로그램 종료\n\n");  printf("메뉴번호 입력>");  select = \_getch() - 48; //내가 입력한거를 안보이게 하기위해서 getch함수를 이용한다. ==> 문자로 받아들여서 -48을 해서 다시 정수로 받아준다.  return select;  }  void sub\_main01(void) {  int c;  while ((c = sub\_menu\_display01()) != 3) {  switch (c) {  case 1: chicken\_burger();  break;  case 2: cheese\_burger();  break;  default:  break;  }  }  }  int sub\_menu\_display01(void) {  int select;  system("cls");  printf("햄버거 만들기\n\n");  printf("1. 치킨버거\n");  printf("2. 치즈버거\n");  printf("3. 메인 메뉴로 이동\n\n");  printf("메뉴번호 입력>");  select = \_getch() - 48;  return select;  }  void chicken\_burger(void) {  system("cls");  printf("치킨버거 만드는 방법\n");  press\_any\_key();  }  void press\_any\_key(void) {  printf("\n\n");  printf("아무키나 누르면 이전 메뉴로...");  \_getch();  }  void cheese\_burger(void)  {  system("cls");  printf("치즈버거 만드는 방법\n");  press\_any\_key();  }  int sub\_menu\_display02(void) {  int select;  system("cls");  printf("스파게티 만들기\n\n");  printf("1. 토마토스파게티\n");  printf("2. 크림스파게티\n");  printf("3. 메인 메뉴로 이동\n\n");  printf("메뉴번호 입력>");  select = \_getch() - 48;  return select;  }  void sub\_main02(void) {  int c;  while ((c = sub\_menu\_display02()) != 3) {  switch (c) {  case 1: tomato\_spa();  break;  case 2: cream\_spa();  break;  default:  break;  }  }  }  void tomato\_spa(void) {  system("cls");  printf("토마토스파게티 만드는 방법\n");  press\_any\_key();  }  void cream\_spa(void) {  system("cls");  printf("크림스파게티 만드는 방법\n");  press\_any\_key();  } |
| * 실행결과   <햄버거 소스 코드 고치기>   |  |  | | --- | --- | |  |  | |

1. 다음과 같은 메인 메뉴를 가지는 프로그램을 작성하시오.

======= 구구단 학습 하기 =======

1. 원하는 단만 보기
2. 구구단 문제 풀기
3. 종료

|  |  |
| --- | --- |
| 1번을 선택하였을 때 출력 예시 | 2번을 선택하였을 때 출력 예시 |
|  |  |

|  |
| --- |
| <소스코드>  #include <stdio.h>  #include <windows.h>  #include <stdlib.h>  #include <time.h>  int menu\_display();  void want\_gogodan();  void view\_gogodan(int);  void press\_any\_key(void);  void problem\_gogodan();  void gotoxy(int, int);  void main()  {  srand((long)time(NULL));  int c;  while ((c = menu\_display()) != 3)  {  switch (c)  {  case 1: want\_gogodan();  break;  case 2: problem\_gogodan();  break;  default:  break;  }  }  }  int menu\_display()  {  int c;  system("cls");  printf("======= 구구단 학습 하기 ======= \n");  printf(" 1. 원하는 단만 보기 \n");  printf(" 2. 구구단 문제 풀기 \n");  printf(" 3. 종료 \n");  c = \_getch() - 48;  return c;  }  void want\_gogodan()  {  int dan;  system("cls");  gotoxy(10, 4);  printf("원하는 단을 입력후 Enter ");  scanf("%d", &dan);  view\_gogodan(dan);  }  void view\_gogodan(int dan)  {  int i;  int x = 10;  int y = 7;  for (i = 1; i < 10; i++)  {  gotoxy(x, y);  printf("%d \* %d = %d \n", dan, i, i\*dan);  y++;  }    press\_any\_key();  }  void press\_any\_key(void) {  printf("\n\n");  gotoxy(10, 25);  printf("아무키나 누르면 이전 메뉴로...");  \_getch();  }  void problem\_gogodan()  {  int i;  int a, b;  int y = 8;  int result[10] = { 0 };  int answer[10] = { 0 };  int cnt = 0;  system("cls");  gotoxy(10, 5);  printf("문제 풀기");  for (i = 0; i < 10; i++)  {  a = rand() % 10 + 1;  b = rand() % 10 + 1;  gotoxy(7, y);  printf("문제 %d > %d \* %d = ? ", i+1, a, b);  result[i] = a \* b;  y++;  }  y = 8;  for (i = 0; i < 10; i++)  {  gotoxy(27, y);  scanf("%d", &answer[i]);  gotoxy(30, y);  if (result[i] == answer[i])  {  printf("O");  cnt += 10;  }  else  {  printf("X");  }  y++;  }  gotoxy(7, y + 2);  printf("당신의 점수는 %d점 입니다.", cnt);  gotoxy(10, y + 4);  printf("수고하셨습니다.");  press\_any\_key();  }  void gotoxy(int x, int y) {  COORD Pos = { x - 1, y - 1 };  SetConsoleCursorPosition(GetStdHandle(STD\_OUTPUT\_HANDLE), Pos);  } |
| <실행결과>   |  |  | | --- | --- | |  |  | |

1. 다음과 같은 메인 메뉴와 서브메뉴를 갖는 프로그램을 작성하시오.
2. 각 메뉴는 ppt 수업자료에 있는 함수를 사용할 것
3. 집합 연산은 다음과 같은 정수형 1차원 배열을 사용하며, 1~20사이의 난수로 초기화 한다. – 함수를 작성하여 처리하시오

int arr1[6]; int arr2[5]

1. 정렬 및 집합 연산 결과출력은 함수를 작성하여 처리하시오
2. 교집합은 본인이 함수를 작성하시오.

======== 메인 메뉴 ===========

1. 정렬
2. 집합 연산

======== 서브메뉴1 ========

1. 선택정렬
2. 버블정렬

======== 서브메뉴2 ========

1. 교집합
2. 합집합
3. 차집합

|  |
| --- |
| * 프로그램 소스   #include <stdio.h>  #include <stdio.h>  int menu\_display();  void submenu01\_display();  void submenu02\_display();  void selectArray();  void bubleArray();  void press\_any\_key();  void go(int a[], int b[], int size\_a, int size\_b, int result[]);  void hap(int a[], int b[], int size\_a, int size\_b, int result[]);  void cha(int a[], int b[], int size\_a, int size\_b, int result[]);  void main()  {  int c;  srand((long)time(NULL));  while ((c = menu\_display()) != 3)  {  switch (c)  {  case 1: submenu01\_display();  break;  case 2: submenu02\_display();  break;  default:  break;  }  }  }  int menu\_display()  {  int select;  system("cls");  printf(" ======== 메인 메뉴 =========== \n");  printf(" 1. 정렬 \n");  printf(" 2. 집합 연산 \n");  select = \_getch() - 48;  return select;  }  void submenu01\_display()  {  int select;  do  {  system("cls");  printf(" ======== 서브메뉴1 ======== \n");  printf(" 1. 선택정렬 \n");  printf(" 2. 버블정렬 \n");  printf(" 3. 메인메뉴로 돌아가기 \n");  select = \_getch() - 48;  if (select == 1)  {  selectArray();  }  else if (select == 2)  {  bubleArray();  }  } while (select != 3);  }  void selectArray()  {  system("cls");  int a[] = { 69, 10, 30, 2, 16, 8, 31, 22 };  int size = sizeof(a) / sizeof(int);  int i, j, t, min, temp;  printf("\n정렬할 원소 : ");  for (t = 0; t < size; t++)  {  printf("%d ", a[t]); // 정렬 전의 원소 출력  }  printf("\n\n<<<<<<<<<< 선택 정렬 수행 >>>>>>>>>>\n");  for (i = 0; i < size - 1; i++)  {  min = i; //기준값 설정  for (j = i + 1; j < size; j++) {  if (a[j] < a[min]) min = j; //가장 작은값이 저장된 index 찾기  }  temp = a[i]; //데이터 교환  a[i] = a[min];  a[min] = temp;  printf("\n%d 단계 : ", i + 1);  for (t = 0; t < size; t++) printf("%3d ", a[t]); //i+1 단계 출력  }  press\_any\_key();  }  void bubleArray()  {  system("cls");  int a[] = { 69, 10, 30, 2, 16, 8, 31, 22 };  int size = sizeof(a) / sizeof(int);  int i, j, t, temp;  printf("\n정렬할 원소 : ");  for (t = 0; t < size; t++)  {  printf("%d ", a[t]);  }  for (i = size - 1; i>0; i--)  {  printf("\n %d 단계>>", size - i);  for (j = 0; j <= i; j++) {  if (a[j - 1]>a[j]) {  temp = a[j - 1]; a[j - 1] = a[j]; a[j] = temp;  }  printf("\n\t");  for (t = 0; t<size; t++) printf("%3d ", a[t]);  }  }  press\_any\_key();  }  void submenu02\_display()  {  int select;  int arr1[6] = { 0 };  int arr2[5] = { 0 };  int i, j;  int result[11] = { 0 };  int size\_a = sizeof(arr1) / sizeof(int);  int size\_b = sizeof(arr2) / sizeof(int);  system("cls");  do  {  for (i = 0; i < 6; i++)  {  arr1[i] = rand() % 20 + 1;  for (j = 0; j < i; j++) // 중복 숫자 가려내기  {  if (arr1[i] == arr1[j])  {  i--;  continue;  }  }  }  for (i = 0; i < 5; i++)  {  arr2[i] = rand() % 20 + 1;  for (j = 0; j < i; j++) // 중복 숫자 가려내기  {  if (arr2[i] == arr2[j])  {  i--;  continue;  }  }  }  system("cls");  printf("======== 서브메뉴2 ======== \n");  printf(" 1. 교집합 \n");  printf(" 2. 합집합 \n");  printf(" 3. 차집합 \n");  printf(" 4. 메인메뉴로 돌아가기");  select = \_getch() - 48;  if (select == 1)  {  go(arr1, arr2, size\_a, size\_b, result);  }  else if (select == 2)  {  hap(arr1, arr2, size\_a, size\_b, result);  }  else if (select == 3)  {  cha(arr1, arr2, size\_a, size\_b, result);  }  } while (select != 4);  }  void go(int a[], int b[], int size\_a, int size\_b, int result[])  {  int i, j, cnt = 0;  system("cls");  printf("========교집합======== \n");  printf("A 집합 : ");  for (i = 0; i < size\_a; i++)  {  printf("%d", a[i]);  if (i < size\_a - 1)  printf(", ");  }  printf("\nB 집합 : ");  for (i = 0; i < size\_b; i++)  {  printf("%d", b[i]);  if (i < size\_b - 1)  printf(", ");  }  for (i = 0; i < size\_a; i++)  {  for (j = 0; j < size\_b; j++)  {  if (a[i] == b[j])  {  result[cnt] = a[i];  cnt++;  }  else  continue;  }  }  printf("\n교집합 결과 : ");  for (i = 0; i < cnt; i++)  {  printf("%d", result[i]);  if (i < cnt - 1)  {  printf(", ");  }  }  press\_any\_key();  }  void hap(int a[], int b[], int size\_a, int size\_b, int result[])  {  int i, j, cnt1 = 0, cnt2 = 0;  system("cls");  printf("========합집합======== \n");  printf("A 집합 : ");  for (i = 0; i < size\_a; i++)  {  printf("%d", a[i]);  if (i < size\_a - 1)  printf(", ");  }  printf("\nB 집합 : ");  for (i = 0; i < size\_b; i++)  {  printf("%d", b[i]);  if (i < size\_b - 1)  printf(", ");  }  for (i = 0; i<size\_a; i++)  result[i] = a[i];  for (i = 0; i<size\_b; i++)  result[size\_a + i] = b[i]; //1단계 완료  for (i = 0; i<size\_a; i++)  {  for (j = 0; j < size\_b; j++)  {  if (a[i] == b[j]) //중복되는것 0으로 대체  {  result[size\_a + j] = 0;  cnt1++;  } //2단계 완료  }  }  for (i = 0; i<size\_b; i++)  {  if (result[size\_a + i] != 0)  {  result[size\_a + cnt2++] = result[size\_a + i]; //3단계 완료  }  }  printf("\n합집합 결과 : ");  for (i = 0; i < size\_a + size\_b - cnt1; i++)  {  printf("%d", result[i]);  if (i < (size\_a + size\_b - cnt1) - 1)  printf(", ");  }  printf("\n");  press\_any\_key();  }  void cha(int a[], int b[], int size\_a, int size\_b, int result[])  {  int i, j, cnt1 = 0, cnt2 = 0;  system("cls");  printf("========차집합======== \n");  printf("A 집합 : ");  for (i = 0; i < size\_a; i++)  {  printf("%d", a[i]);  if (i < size\_a - 1)  printf(", ");  }  printf("\nB 집합 : ");  for (i = 0; i < size\_b; i++)  {  printf("%d", b[i]);  if (i < size\_b - 1)  printf(", ");  }  for (i = 0; i<size\_a; i++)  result[i] = a[i]; //1단계, 배열 a의 원소를 result에 저장  for (i = 0; i<size\_a; i++) {  for (j = 0; j<size\_b; j++)  if (a[i] == b[j])  {  result[i] = 0;  cnt1++;  } //2단계,배열 a와 b에 동일한 원소가 있으면 0 저장  }  for (i = 0; i<size\_a; i++) {  if (result[i] != 0)  result[cnt2++] = result[i]; //3단계  }  printf("\n차집합 결과 : ");  for (i = 0; i < cnt2; i++)  {  if (result[i] != 0)  {  printf("%d", result[i]);  if (i < cnt2 - 1)  printf(", ");  }  else  break;  }  printf("\n");  press\_any\_key();  }  void press\_any\_key(void) {  printf("\n\n");  printf("아무키나 누르면 이전 메뉴로...");  \_getch();  } |
| * 실행결과 |