|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **C1\_실습과제\_9주** | **이름 :** |  | **학번 :** |  |

* **강의 내용**

1. 빈도수 계산

|  |
| --- |
| #include <stdio.h>  int main(){  int ary[]={2,8,15,1,8,10,5,19,19,3,5,6,6,2,8,2,12,16,3,8,17,  12,5,3,14,13,3,2,17,19,16,8,7,12,19,10,13,8,20,  16,15,4,12,3,14,14,5,2,12,14,9,8,5,3,18,18,20,4};  int i, j, size;  **// 빈도수를 계산하여 저장할 배열, 초기화 없으면 쓰레기 값으로 채워짐**  int count[20]={0};  size=sizeof(ary)/sizeof(ary[0]); **//배열 크기 계산**  for(i=0; i<20; i++){ **//배열에 저장된 값이 1~20**  for(j=0; j<size; j++){  **//배열에 저장된 숫자(ary[j])와 일치하는 인덱스(i+1)에 저장된 값을 1씩 증가**  if(ary[j]==i+1) count[i]++; **//각 숫자에 대한 빈도수 계산**  }  }  return 0;  } |

1. 크기순으로 정렬

|  |
| --- |
| int list[]={4, 2, 6, 12, 1, 50};  int temp, least, size, i, j;  size=sizeof(list)/sizeof(list[0]); **//배열 크기 계산**  for (i = 0; i < size; i++) {  least = i;  for (j = i + 1; j < size; j++){  if (list[j] < list[least]) **// 최소값 탐색**  least = j; **// 최소값을 가지는 배열 원소의 색인을 저장**  }  temp = list[i]; **// i번째 원소와 least 위치의 원소를 교환**  list[i] = list[least];  list[least] = temp;}  } |

❑ 프로그래밍 과제

1. “빈도수 계산” 예제에서 빈도수를 계산하는 함수를 작성하시오. 단, 함수 호출 시 빈도수 계산을 위한 배열과, 빈도수를 저장할 배열을 형식매개변수로 전달

#include <stdio.h>

int frequency(int arr[],int count[],int size);

int main(void){

int size;

int ary[]={2,8,15,1,8,10,5,19,19,3,5,6,6,2,8,2,12,16,3,8,17,

12,5,3,14,13,3,2,17,19,16,8,7,12,19,10,13,8,20,

16,15,4,12,3,14,14,5,2,12,14,9,8,5,3,18,18,20,4};

int count[20]={0};

size = sizeof(ary)/sizeof(ary[0]);

frequency(ary,count,size);

return 0;

}

int frequency(int ary[], int count[],int size){

int i, j,num;

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

for(j=0; j<size; j++){

if(ary[j]==i+1) count[i]++;

}

}

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

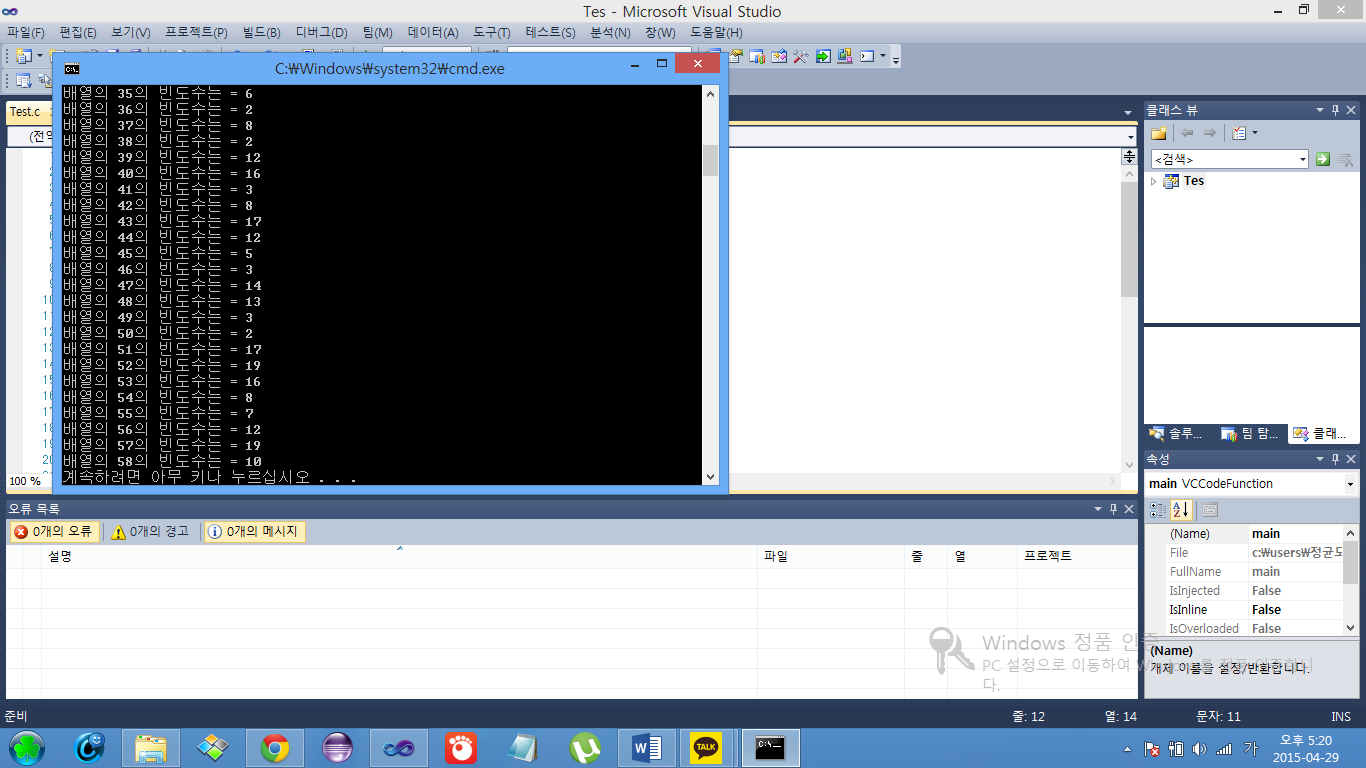
for(num=0;num<size;num++){

printf("배열의 %d의 빈도수는 = %d\n",num+1,count[num]);

}

return 0;

}



1. 1부터 20사이의 수로 생성된 난수 30개에서 가장 많은 빈도 수와 숫자가 무엇인지 출력하는 프로그램을 작성하시오.

* 빈도수 계산은 1에서 만들어진 함수를 사용
* 가장 많은 빈도수를 계산하여 출력하는 함수를 정의하여 사용

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#define MAX 20

int frequency(int arr[],int count[],int size);

int main(void){

int size,i;

int ary[30];

int count[20]={0};

srand((long)time(NULL));

size = sizeof(ary)/sizeof(ary[0]);

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

ary[i]=rand()%(MAX+1);

}

frequency(ary,count,size);

return 0;

}

int frequency(int ary[], int count[],int size){

int i, j,num,v;

int max=ary[1];

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

for(j=0; j<size; j++){

if(ary[j]==i+1) count[i]++;

}

}

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

for(num=0;num<size;num++){

printf("배열의 %d의 빈도수는 = %d\n",num+1,count[num]);

}

for(v=1;v<size;v++){

if(max<ary[v]){

max=ary[v];

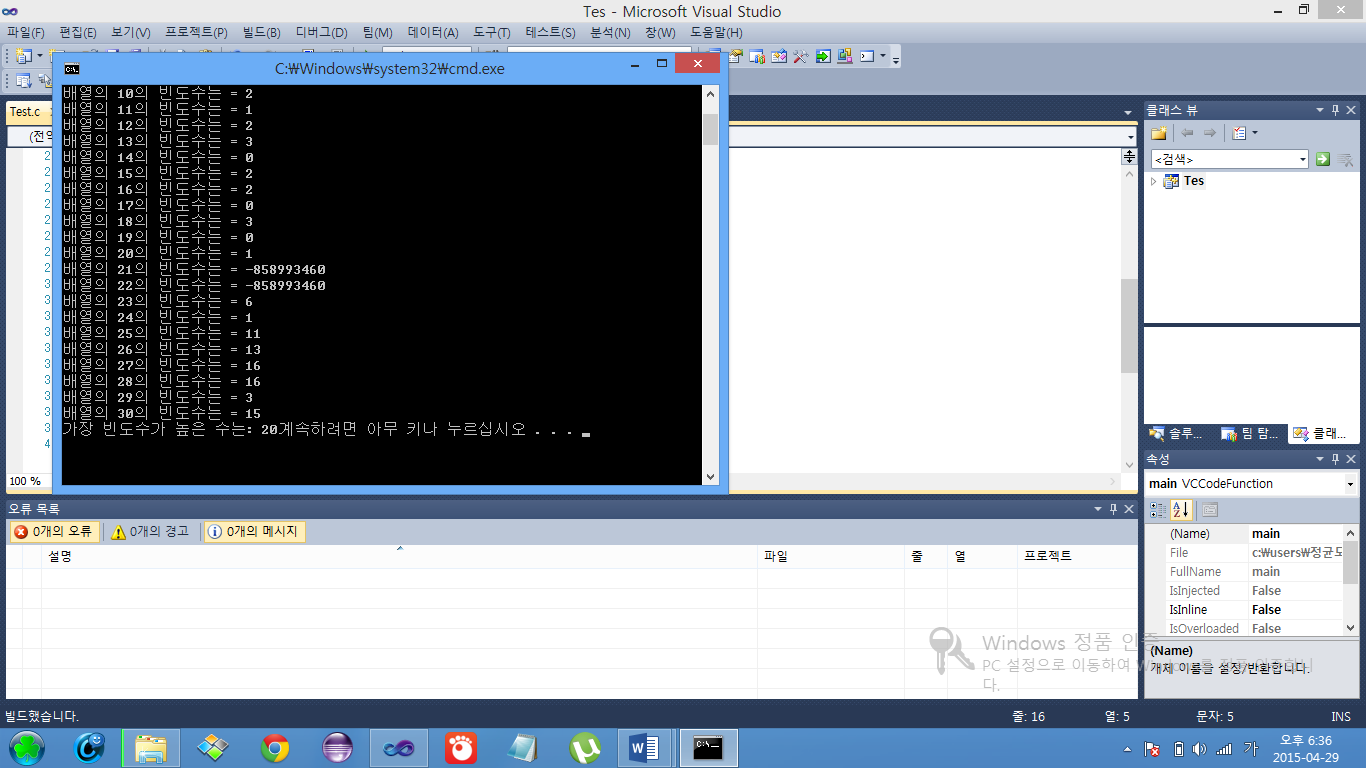
}

}

printf("가장 빈도수가 높은 수는: %d",max);

return 0;

}



1. 반복문을 이용하여 다음과 같은 수를 배열에 10개에 순서대로 저장하고 이 값을 출력하는 프로그램을 작성하시오.

1/(2\*3), 1/(3\*4), 1/(4\*5), …, 1/(11\*12)

#include <stdio.h>

int main(void){

double size;

int i,j;

double ary[10];

double fh=2;

size = sizeof(ary)/sizeof(ary[0]);

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

ary[i]=1/(fh)\*(fh+1);

fh++;

}

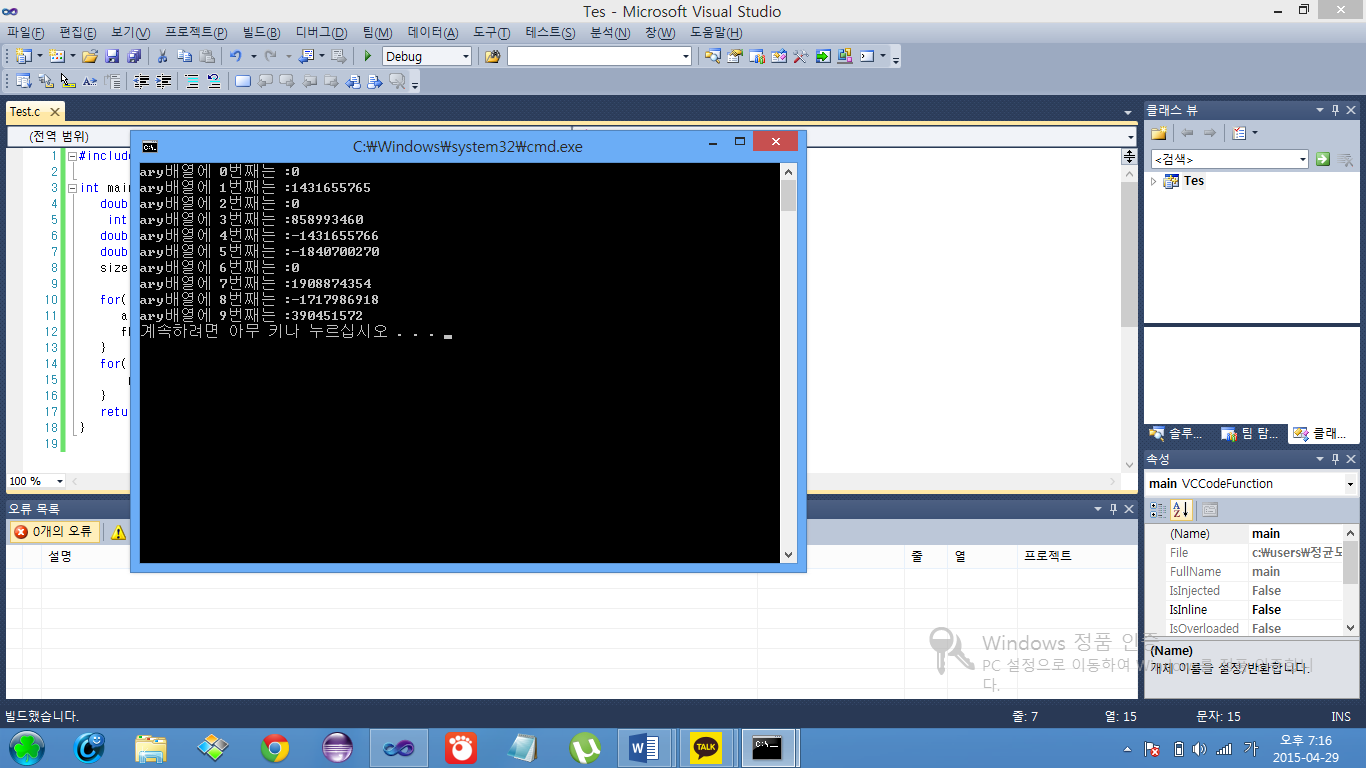
for(j=0;j<size;j++){

printf("ary배열에 %d번째는 :%d\n",j,ary[j]);

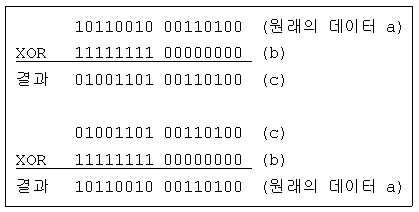
}

return 0;

}



1. XOR연산을 이용하여 암호화하고 변환된 암호문을 원래의 값으로 복원하는 프로그램을 작성하시오. 프로그램은 다음과 같은 함수로 구성된다. 함수 호출 시 마스크 비트 값과, 배열, 배열 크기를 전달한다. 마스크 비트 값과 문자 배열 값은 임의로 처리할 것



void XOR\_decryption(int key, char [], int);

#include <stdio.h>

void XOR\_decryption(int, char[], int );

int main()

{

int size, key = 10, i;

char c[10]={0};

size = sizeof(c);

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

{

scanf("%c",&c[i]);

}

XOR\_decryption(key, c, size);

return 0;

}

void XOR\_decryption(int ky, char a[], int si)

{

int i;

char re[10];

printf("원래의 값: \n");

for(i=0; i<si; i++)

{

printf("%c",a[i]);

}

printf("\n");

printf("key값과 xor연산 된 값: \n");

for(i=0; i<si; i++)

{

re[i]=a[i]^ky;

printf("%c",re[i]);

}

printf("\n");

printf("연산된 값에 다시 key값을 xor연산 한 값: \n");

for(i=0; i<si; i++)

{

printf("%c",re[i]^ky);

}

printf("\n");

}

