2022 CAT-CERT 스터디 보고서

컴퓨터정보공학부 202221168 김지혜

2022.05.07#C\_contest

#2379

|  |
| --- |
| #include<stdio.h>  #include<string.h>  int main() {  int num, rest;  int i = 0;  int answer[100] = { 0, };  scanf("%d", &num);  while (1) {  rest = num % 4;  if (rest == 0) {  num = num - 4;  answer[i] = 9;  }  else if(rest==1)  answer[i] = 2;  else if(rest==2)  answer[i] = 3;  else if(rest==3)  answer[i] = 7;  i++;  num = num / 4;  if (num == 0) {  break;  }  }  for (int j = i-1; j >=0; j--) {  printf("%d", answer[j]);  }  } |

#crypto

|  |
| --- |
| #include<stdio.h>  #include<string.h>  int main() {  char select[2] = { 0, };  int num, rest;  char hex[20] = { 0, };  char hex1[20] = { 0, };  int idx = 0;  char str[101] = { 0, };  int answer = 0;  int number;  int len = 0;  int len2 = 0;  scanf("%c", &select);  scanf("%d", &num);  scanf(" %[^\n]s", str);  while (1) { //16진수로  rest = num % 16;  if (rest < 10) {  hex[idx] = 48 + rest; //숫자 48~57  }  else {  hex[idx] = (65 + (rest - 10)); //문자 65~70  }  num = num / 16;  idx++;  if (num == 0)  break;  }  //배열값을 10진수로  if ((int)hex[0] < 58) {  number = ((int)hex[0] - 48) \* 16;  }  else {  number = ((int)hex[0] - 55) \* 16;  }  if (hex[1] != NULL) {  if ((int)hex[1] < 58) {  number = number + ((int)hex[1] - 48);  }  else {  number = number + ((int)hex[1] - 55);  }  }  //암호화 복호화  len = strlen(str);  if (strcmp(select, "E") == 0) {  for (int i = 0; i<len; i++) {  idx = 0;  answer = (int)str[i] ^ number;  while (1) {  rest = answer % 16;  if (rest < 10) {  hex1[idx] = 48 + rest; //숫자 48~57  }  else {  hex1[idx] = (97 + (rest - 10)); //문자 65~70  }  answer = answer / 16;  idx++;  if (answer == 0)  break;  }  len2 = strlen(hex1);  for (int j = idx - 1; j >= 0; j--) {  if (len2 == 1) {  printf("0%c", hex1[j]);  }  else {  printf("%c", hex1[j]);  hex1[j] = 0;  }  }  }  }  else if (strcmp(select, "D") == 0) {  for (int i = 0; i<len; i++) {  answer = (int)str[i] ^ number;  printf("%s", &answer);  }  }  } |

#CATPUTER 1

|  |
| --- |
| #include<stdio.h>  #include<string.h>  int main() {  char str[5] = { 0, };  unsigned num;  int count = 0;  while (1) {  scanf("%s", &str);  if (strcmp(str, "RURI") == 0) {  scanf("%d", &num);  count++;  }  else if (strcmp(str, "MIRU") == 0)  count--;  else if (strcmp(str, "END") == 0)  break;  }  if (count <= 0)  printf("FALSE\n");  else {  printf("TRUE\n");  }  } |

#CATPUTER 2

|  |
| --- |
| #include<stdio.h>  #include<string.h>  int main() {  char str[5] = { 0, };  unsigned int cnt[100] = { 0, };  unsigned int num;  int k = 0;  int count = 0;  while (1) {  scanf("%s", &str);  if (strcmp(str, "RURI") == 0) {  scanf("%d", &num);  cnt[k] = num;  k++;    }  else if (strcmp(str, "MIRU") == 0) {  cnt[k-1] = 0;  k--;  }  else if (strcmp(str, "END") == 0)  break;  }  if (k > 0) {  for (int i = k; i > 0; i--) {  printf("%u\n", cnt[i - 1]);  }  }  else {  printf("FALSE\n");  }  } |

#CATPUTER 3

|  |
| --- |
| #include<stdio.h>  #include<string.h>  int main() {  char str[5] = { 0, };  unsigned int cnt[100] = { 0, };  unsigned int num;  int select;  int k = 0;  int count = 0;  while (1) {  scanf("%s", &str);  if (strcmp(str, "RURI") == 0) {  scanf("%d", &num);  cnt[k] = num;  k++;    }  else if (strcmp(str, "MIRU") == 0) {  cnt[k-1] = 0;  k--;  }  else if (strcmp(str, "HYUN") == 0) {  scanf("%d", &select);  if (select == 0) {  cnt[k - 2] = cnt[k - 1] + cnt[k - 2];  k--;  }  else if (select == 1) {  cnt[k - 2] = cnt[k - 2] - cnt[k - 1];  k--;  }  else if (select == 2) {  cnt[k - 2] = cnt[k - 2] \* cnt[k - 1];  k--;  }  else if (select == 3) {  cnt[k - 2] = cnt[k - 2] & cnt[k - 1];  k--;  }  else if (select == 4) {  cnt[k - 2] = cnt[k - 2] | cnt[k - 1];  k--;  }  else if (select == 5) {  cnt[k - 2] = cnt[k - 2] ^ cnt[k - 1];  k--;  }  }  else if (strcmp(str, "END") == 0)  break;  }  if (k > 0) {  for (int i = k; i > 0; i--) {  printf("%u\n", cnt[i - 1]);  }  }  else {  printf("FALSE\n");  }  } |

#CATPUTER 4

|  |
| --- |
| #include<stdio.h>  #include<string.h>  int main() {  int i = 0;  char name[100][5] = { 0, };  int num[100] = { 0, };  unsigned int cnt[100] = { 0, };  int k = 0;  int j = 0;  while (1) {  scanf("%s",name+i);  if (strcmp(name[i],"MIRU") == 0 || strcmp(name[i],"MEOW") == 0 || strcmp(name[i],"END") == 0) {  num[i] = 0;  }  else {  scanf("%d", &num[i]);  }  if (strcmp(name[i], "END") == 0) {  break;  }  i++;  }  while (strcmp(name[j], "END") != 0) {  if (strcmp(name[j], "RURI") == 0) {  cnt[k] = num[j];  k++;  j++;  }  if (strcmp(name[j], "MIRU") == 0) {  cnt[k - 1] = 0;  j++;  k--;  }  if (strcmp(name[j], "HYUN") == 0) {  if (num[j] == 0) {  cnt[k - 2] = cnt[k - 1] + cnt[k - 2];  k--;  }  else if (num[j] == 1) {  cnt[k - 2] = cnt[k - 2] - cnt[k - 1];  k--;  }  else if (num[j] == 2) {  cnt[k - 2] = cnt[k - 2] \* cnt[k - 1];  k--;  }  else if (num[j] == 3) {  cnt[k - 2] = cnt[k - 2] & cnt[k - 1];  k--;  }  else if (num[j] == 4) {  cnt[k - 2] = cnt[k - 2] | cnt[k - 1];  k--;  }  else if (num[j] == 5) {  cnt[k - 2] = cnt[k - 2] ^ cnt[k - 1];  k--;  }  j++;  }  if (strcmp(name[j], "TIME") == 0) {  j += num[j];  }  if (strcmp(name[j], "MEOW") == 0) {  if (k > 0) {  printf("%u\n", cnt[k - 1]);  j++;  k--;  }  else {  printf("FALSE\n");  }  }  }  } |

#game\_world

|  |
| --- |
| #include<stdio.h>  #include<string.h>  int main() {  int num1, num2;  scanf("%d %d", &num1, &num2);  int arr[11][11] = { 0, };  char direct[11] = { 0, };  int sum = 0;  int n = 0;  scanf("%s",&direct);  int len = strlen(direct);  for (int i = 0; i < num1; i++) {  for (int j = 0; j < num2; j++) {  scanf("%d", &(arr[i][j]));  }  }  sum = arr[num1-1][0];  int num = num1 - 1;  for (int k = 0; k < len; k++) {  if (direct[k] == 'w') {  if (num == 0) {  sum += arr[num][n];  }  else {  sum += arr[num -1][n];  num--;  }  }  if (direct[k] == 'a') {  if (n == 0) {  sum += arr[num][n];  }  else {  sum += arr[num][n - 1];  n--;  }  }  if (direct[k] == 's') {  if (num == num2-1) {  sum += arr[num][n];  }  else {  sum += arr[num + 1][n];  num++;  }  }  if (direct[k] == 'd') {  if (n == num2-1) {  sum += arr[num][n];  }  else {  sum += arr[num][n + 1];  n++;  }  }  }  printf("%d\n", sum);  } |

#mi\_ru\_ri

|  |
| --- |
| #include<stdio.h>  int main() {  int num;  char arr[101][101] = { 0, };    scanf("%d", &num);  for (int i = 0; i < num; i++) {  for (int j = 0; j < num; j++) {  scanf(" %c", &arr[i][j]);  }  }    int n = num - 1;  int dx[5] = { -1, 1, 0, 0 };  int dy[5] = { 0, 0, -1, 1 };  for (int i = 0; i < num; i++) {  for (int j = 0; j < num; j++) {  int newI, newJ;  if (arr[i][j] == 'Y') {  // M  for (int idx = 0; idx < 4; idx++) {  newI = i + dx[idx];  newJ = j + dy[idx];  if (newI < 0 || newI == num || newJ < 0 || newJ == num)  continue;  arr[newI][newJ] = 'M';  }  //R  if (n - j < j) {  if (n - i < i) {  arr[0][0] = 'R';  }  else if (n - i > i) {  arr[n][0] = 'R';  }  else if (n - i == i) {  arr[0][0] = 'R';  arr[n][0] = 'R';  }  }  else if (n - j > j) {  if (n - i < i)  arr[0][n] = 'R';  else if (n - i > i) {  arr[n][n] = 'R';  }  else if (n - i == i) {  arr[0][n] = 'R';  arr[n][n] = 'R';  }  }  else if (n - j == j) {  if (i > n - i) {  arr[0][0] = 'R';  arr[0][n] = 'R';  }  else if (i < n - i) {  arr[n][0] = 'R';  arr[n][n] = 'R';  }  else if (i == n - i) {  arr[0][0] = 'R';  arr[0][n] = 'R';  arr[n][0] = 'R';  arr[n][n] = 'R';  }  }  }  }  }  for (int k = 0; k < num; k++) {  for (int s = 0; s < num; s++) {  printf("%c", arr[k][s]);  printf(" ");  }  printf("\n");  }  } |

#hate\_summer

|  |
| --- |
| #include<stdio.h>  int main() {  char dest[5] = { 0, };    scanf("%s", dest);  if (dest[0] == 'K') {  if (dest[1] != '1') {  printf("K1F-E-K%cF-", dest[1]);  }  else {  printf("K1F-");  }  }  else {  printf("K1F-E-K4F-");  if (dest[1] == '1') {  printf("%c1F-", dest[0]);  }  else {  printf("M1F-E-");  if (dest[0] == 'M') {  printf("M%cF-", dest[1]);  }  else if (dest[0] == 'N') {  printf("M%cF-N%cF-", dest[1], dest[1]);  }  else if (dest[0] == 'V') {  printf("M%cF-N%cF-V%cF-", dest[1] + 1, dest[1] + 1, dest[1]);  }  else if (dest[0] == 'B') {  if (dest[1] == '3')  printf("M%cF-N%cF-V%cF-B%cF-", dest[1] + 1, dest[1] + 1, dest[1], dest[1]);  else if (dest[1] == '2')  printf("M%cF-N%cF-V%cF-B%cF-B%cF-", dest[1] + 2, dest[1] + 2, dest[1] + 1, dest[1] + 1, dest[1]);  }  else if (dest[0] == 'D') {  printf("M4F-N4F-D2F-");  if (dest[1] != '2')  printf("E-D%cF-", dest[1]);    }  }  }  printf("ARRIVE\n");  } |

#mystery\_sign

|  |
| --- |
| #include<stdio.h>  int main() {  char arr[100] = { 0, };  char arr2[100] = { 0, };  char num[10] = { 0, };  int f, s;  int res;  scanf("%d %d %s %s", &f, &s, arr, arr2);  for (int i = 0; i < 10; i++) {  for (int j = 0; j < f; j++) {  if ((int)arr[j] == i + 48) {  num[i]++;  }  }  for (int j = 0; j < s; j++) {  if ((int)arr2[j] == i + 48) {  num[i]++;  }  }  }  printf("%s ? %s = ", arr, arr2);  for (int j = 0; j < f; j++) {  res = (int)arr[j] - 48;  if (num[res] != 0) {  printf("%c%d", arr[j], num[res]);  num[res] = 0;  }  }  for (int j = 0; j < s; j++) {  res = (int)arr2[j] - 48;  if (num[res] != 0) {  printf("%c%d", arr2[j], num[res]);  num[res] = 0;  }  }  } |

#bubble

|  |
| --- |
| #include<stdio.h>  #include<string.h>  int main() {  char arr[100] = { 0, };  char arr2[100] = { 0, };  scanf("%[^\n]s", arr);  int j = 0;  int len = strlen(arr);  for (int i = 0; i < len; i++) {  if (arr[i] != ' ') {  if ((int)arr[i] > 64 && (int)arr[i] < 91) {  arr2[j] = arr[i];  j++;  }  else if ((int)arr[i] > 96 && (int)arr[i] < 123) {  arr2[j] = arr[i];  j++;  }  }  else {  arr2[j] = arr[i];  j++;  }  }  printf("%s\n", arr2);  } |

#broken\_keyboard

|  |
| --- |
| #include<stdio.h>  #include<string.h>  int main() {  char arr[100] = { 0, };  char arr2[100] = { 0, };  scanf("%[^\n]s", arr);  int j = 0;  int len = strlen(arr);  for (int i = 0; i < len; i++) {  if (arr[i] != ' ') {  if ((int)arr[i] > 64 && (int)arr[i] < 91) {  arr2[j] = arr[i];  j++;  }  else if ((int)arr[i] > 96 && (int)arr[i] < 123) {  arr2[j] = arr[i];  j++;  }  }  else {  arr2[j] = arr[i];  j++;  }  }  printf("%s\n", arr2);  } |

#parity\_bit

소스코드

|  |
| --- |
| #include<stdio.h>  #include<string.h>  int main() {  char num[32] = { 0, };  scanf("%s", &num);  int len;  len = strlen(num);  int n = 1;  int sum = 0;  int count = 0;  for (int j = 0; j < len; j++) {  if (num[j] == '1')  count++;  }    for (int i = len-1; i >= 0; i--) {  sum += ((int)(num[i])-48) \* n;  n \*= 2;  }    if (sum==0) {  printf("0x%X\n", sum);  }  else{  if (count % 2 != 0)  printf("0x%X\n", (sum \* 2) + 1);  else  printf("0x%X\n", sum);  }  } |

#Basic

컴파일러 :

프로그래밍 언어로 쓰여진 것을 컴퓨터언어로 옮기는 일종의 번역 프로그램 ; 언어에 따라 다양한 컴파일러 종류가 존재함

원래의 문서 -> 소스코드

출력된 문서 -> 목적코드

컴파일 과정 :

1. 전처리기 -> ‘#include 등 ‘#’이 붙은 전처리기 구문 처리
2. 컴파일러로 컴파일
3. 어셈블러 -> 완전히 기계어로 바꾸어줌
4. 링커 -> 여러 개의 오브젝트파일이나 라이브러리를 합침

|  |  |
| --- | --- |
| 디버그 모드 | 릴리즈 모드 |
| : 디버깅에 적합한 모드   1. 코드 최적화하지 않음 2. 실행파일의 크기가 큼 3. 코드 실행 속도 느림 4. 메모리 사용량 많음 5. 바이너리에 디버깅에 필요한 정보 포함 6. 컴파일 속도는 빠름   ->개발 진행중인 프로젝트에서 사용 | : 배포에 적합한 모드   1. 코드 최적화 수행 2. 실행파일 크기 작음 3. 코드 실행속도 빠름 4. 메모리 사용량 적음 5. 디버깅에 필요한 정보 포함되지 않음 6. 컴파일 속도가 느림   (최적화 과정 포함돼서)  ->실제 사용자에게 전달될 때 사용 |

#<String.h> 헤더 파일 함수

1. Strlen ; strlen(str);

-> ‘\0’문자가 나올 때 까지의 문자들의 개수 ; 문자길이

2. strcmp ; strcmp(str1,str2);

-> 두 문자열을 비교하여 같으면 ‘0’을 반환함

\*\*Str1>str2 -> 양수 반환

str1<str2 -> 음수 반환

1. strcpy ; strcpy(dest, origin);

-> origin에 있는 문자열을 dest 에 복사함

(‘\0’값까지 복사함)

1. atoi ; atoi(str);

->문자열을 정수값으로 변환함

(소수점이나 지수를 인식하지 못하고, 널 문자일 경우 입력 문자열 읽는 것을 종료함)

1. itoa ; itao(value,str,base);

->숫자를 문자로 변환해 주는 함수

변환할 값의 진수(base)를 선택하고 그 값을 저장할 변수(str)을 입력한다.

=>라이브러리 함수가 아니라 일부만 지원함

#리틀엔디안

:최하위 바이트부터 차례대로 저장함

0x12345678

=> 0x78563412

데이터의 상위 바이트가 주소값이 큰 곳에 저장이 됨

->인텔 계열은 리틀엔디안 방식으로 데이터 저장

#빅 엔디안

:최 상위 바이트부터 차례대로 저장함

0x12345678

=>0x12345678

데이터의 상위 바이트가 주소값이 작은 곳에 저장이 됨

->네트워크 상 데이터를 전송할 때는 데이터의 배열을 빅엔디안으로 변경해서 송신해야 올바른 데이터 처리가 가능함

Code -> 실행할 프로그램의 코드가 저장됨

Stack-> 지역변수가 저장됨

함수의 호출과 관계되는 지역변수와 매개변수가 저장되는 영역으로 함수의 호출과 함께 할당되고 완료되면 소멸됨

Heap-> 동적할당변수가 저장됨(malloc)

사용자가 직접관리 할 수 있고, 해야만 하는 영역으로 사용자에 의해 메모리 공간이 동적으로 할당되고 해제됨

Data->

1.BSS-> 초기화 되지 않은 전역,정적변수가 저장됨

1. data -> 초기화 된 전역, 정적 변수가 저장됨
2. rodata-> 읽기 전용의 데이터가 저장됨