

제17회 교원컴퓨터프로그래밍 경진대회(예선)

2020. 9. 5.(토) 수험번호() 소속() 이름() 경상남도교육청

1. 네트워크 기반 공격에 대한 설명으로 옳은 것은? (3.8점)

- ① 스푸핑 - 네트워크 상의 데이터를 감청하는 기법
- ② 스니핑 - A와 B 사이의 통신에 끼어들어 패킷을 위조하는 기법
- ③ 분산 반사 서비스 거부 - 정상적인 서비스 부하량을 늘리는 기법
- ④ 서비스 거부 - 외부에서 물리적인 장치명, 운영체제 등의 정보를 알아내는 행위
- ⑤ 네트워크 스캐닝 - 네트워크에 연결된 서버나 클라이언트의 개방 포트를 조사하는 행위

2. 메모리 주소 A, B, C, D에 1, 2, 3, 4가 순서대로 저장되어 있다. 아래의 명령어 실행 후 메모리 주소 E에 저장되는 결과는? (3.7점)

```
MOVE R1, B
MOVE R2, C
MUL R1, R2
ADD R1, A
ADD R1, D
MOV E, R1
```

- ① 8
- ② 9
- ③ 10
- ④ 11
- ⑤ 12

3. 캐시기억장치 접근시간이 100ms이고 주기억장치 접근시간이 200ms이다. 캐시 적중률이 40%일 때의 캐시 기반 평균 메모리 접근시간을 s1, 캐시 적중률이 80%일 때의 캐시 기반 평균 메모리 접근시간을 s2라고 할 때, s1과 s2의 값을 순서대로 나타낸 것은? (3.8점)

- ① 180ms, 120ms
- ② 160ms, 120ms
- ③ 120ms, 160ms
- ④ 140ms, 180ms
- ⑤ 180ms, 140ms

4. 인터럽트에 대한 설명으로 옳은 것은? (4점)

- ① 오버플로우나 0으로 나누기가 발생한 경우 외부 인터럽트가 발생한다.
- ② 프로그램 처리 중 명령의 요청에 의해 내부 인터럽트가 발생한다.
- ③ 인터럽트의 종류는 외부 인터럽트와 내부 인터럽트로만 나뉜다.
- ④ 프로그램 실행 중 예상치 않은 상황으로 인해 실행 중인 작업을 중단하고 발생한 상황을 처리한다.
- ⑤ 하드웨어 방식의 인터럽트 우선순위 판별은 인터럽트 요청 플래그를 우선순위 별로 검사하여 서비스 루틴을 수행한다.

5. 다음 식을 불대수의 기본 정리를 이용하여 최대로 간소화한 것은? (3.9점)

$$\overline{(A+B+C)} + B \cdot C + A \cdot \overline{B} \cdot \overline{C} + \overline{(B+C)}$$

- ① $B \cdot C + \overline{B}$
- ② $\overline{A} \cdot \overline{B} \cdot \overline{C} + B \cdot C$
- ③ $\overline{B} \cdot \overline{C} + B$
- ④ $\overline{B} \cdot \overline{C} + \overline{A} \cdot B + A \cdot B$
- ⑤ $A \cdot \overline{C} + B$

6. 가상기억장치 관리기법 중 앞으로 가장 오랫동안 사용하지 않을 페이지를 교체하는 것은? (3.6점)

- ① FIFO(First In First Out)
- ② LFU(Least Frequently Used)
- ③ LRU(Least Recently Used)
- ④ NUR(Not Used Recently)
- ⑤ 최적교체

7. 패리티 검사 방식 중 수직 중복 검사 방식을 적용했을 때, 빈칸 ㉠에 들어갈 패리티 비트로 옳은 것은? (3.8점)

㉠							
0	1	0	1	1	1	0	1
0	0	0	0	1	1	1	0
1	0	1	0	1	0	1	0
1	1	1	0	0	1	0	0
0	0	0	0	1	0	1	0
0	0	0	1	0	0	1	0
1	0	1	0	1	0	1	1

- ① 10101110 ② 01011110 ③ 10101010
 ④ 01011011 ⑤ 01010000

8. 명제인 것만을 <보기>에서 있는 대로 고른 것은? (3.7점)

<보기>

ㄱ. 모든 3의 배수는 홀수이다.
 ㄴ. 배구 선수들은 키가 크다.
 ㄷ. $x^3 + 3x^2 + 3x + 1 = 0$

- ① ㄱ ② ㄴ ③ ㄱ, ㄷ ④ ㄴ, ㄷ ⑤ ㄱ, ㄴ, ㄷ

9. 다음 수식의 계산 결과는? (3.6점)

$1110_{(2)} + 15_{(16)} + 13_{(8)}$

- ① 40 ② 42 ③ 44 ④ 46 ⑤ 48

10. 2의 보수 표현에 대한 설명으로 옳은 것을 <보기>에서 있는 대로 고른 것은? (3.9점)

————— <보기> —————

ㄱ. 음수의 최상위비트는 1이다. ㄴ. 1의 보수에 1을 더한 수이다. ㄷ. 어떤 수 n 과의 합이 1이 되는 수이다. ㄹ. 30의 2의 보수 표현은 $11100001_{(2)}$ 이다.

- ① ㄱ, ㄴ ② ㄱ, ㄷ ③ ㄴ, ㄷ ④ ㄴ, ㄹ ⑤ ㄷ, ㄹ

11. 집합 $A=\{1,2,3,4,6,12\}$ 에 관한 관계 $R=\{(a,b) \mid (a \bmod b) \in A, a,b \in A\}$ 로 정의되어 있다. 다음에서 관계 R의 성질에 대해서 옳은 것만을 <보기>에서 있는 대로 고른 것은? (4.3점)

————— <보기> —————

ㄱ. 추이관계	ㄴ. 비반사관계	ㄷ. 반대칭관계
---------	----------	----------

- ① ㄱ ② ㄴ ③ ㄱ, ㄷ ④ ㄴ, ㄷ ⑤ ㄱ, ㄴ, ㄷ

12. $n!$ 은 1부터 n 까지의 정수를 연속적으로 곱한 값이다. $20!$ 의 오른쪽 끝에 연속되는 0의 개수는? (4점)

$n!$ 예시	오른쪽 끝에 연속되는 0의 개수
$3! = 3 \times 2 \times 1 = 6$	0
$6! = 6 \times 5 \times 4 \times 3 \times 2 \times 1 = 720$	1
$12! = 12 \times 11 \times 10 \times 9 \times 8 \times 7 \times 6 \times 5 \times 4 \times 3 \times 2 \times 1 = 479,001,600$	2

- ① 2 ② 3 ③ 4 ④ 6 ⑤ 12

13. 다음은 스택의 push와 pop 연산을 구현한 것이다. 밑줄 친 ㉠, ㉡에 들어갈 내용으로 옳은 것은? (4점)

```

typedef struct{
    int stack[10];
    int top;
} Stack;
void init(Stack *s) {
    s->top = -1;
}
void push(Stack *s, int item) {
    if( s->top < 10 ) {
        s->stack[          ㉠          ] = item;
    }
}
int pop(Stack *s) {
    if(          ㉡          ) {
        return s->stack[(s->top)--];
    }
}

```

- | | ㉠ | ㉡ |
|---|------------|-------------|
| ① | s->top | s->top != 0 |
| ② | ++(s->top) | s->top > 0 |
| ③ | ++(s->top) | s->top > -1 |
| ④ | (s->top)++ | s->top > -1 |
| ⑤ | (s->top)++ | s->top != 0 |

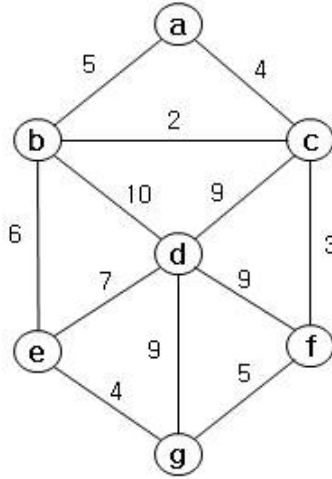
14. 트리에 대해서 사용하는 몇 가지 표기법을 아래와 같이 정의한다.

- height(n): 노드 n을 루트로 하는 트리의 높이
- count(n): 노드 n을 루트로 하는 트리에 있는 모든 노드의 수
- n.left: 노드 n의 왼쪽 자식노드
- n.right: 노드 n의 오른쪽 자식노드

다음에서 완전이진트리에 관한 식으로 옳은 것은? (이 트리의 루트 노드는 r이다.) (4.1점)

- ① height(r.left) ≤ height(r.right)
- ② count(r) ≤ 2^{height(r)} - 1
- ③ count(r.left) ≤ count(r.right)
- ④ count(r) ≤ 2 × count(r.left)
- ⑤ height(r) = height(r.right) + 1

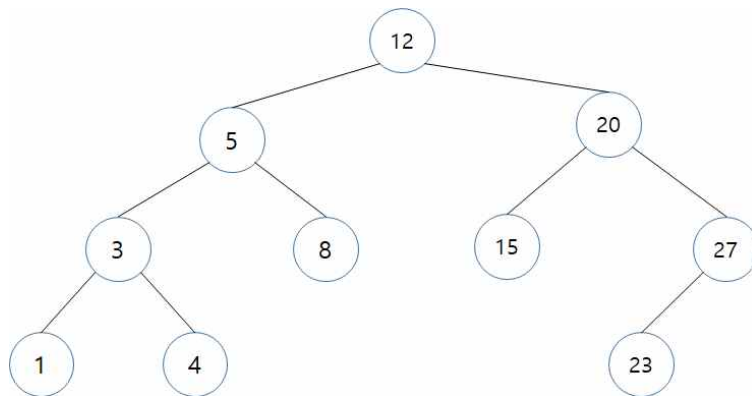
15. 최소비용신장트리를 구하는 Prim 알고리즘은 시작 정점에서부터 출발하여 신장트리 집합을 단계적으로 확장해나가는 방법이다. 시작 단계에서는 시작 정점만이 신장트리 집합에 포함되고, 앞 단계에서 만들어진 신장트리 집합에 인접한 정점들 중에서 최저 간선으로 연결된 정점을 선택하여 트리를 확장한다. 이 과정은 n개의 정점을 가진 트리가 n-1개의 간선을 가질 때까지 계속된다.



위의 그래프에 대하여 최소비용신장트리를 Prim의 방법으로 구할 때, 6번째로 선택되는 정점은? (단, 시작 정점 a가 첫 번째로 선택되는 정점이다.) (3.9점)

- ① b ② c ③ d ④ e ⑤ f

16. 다음 이진탐색트리에 11, 13, 14가 순서대로 삽입된 후 15가 삭제되었다. 이때 만들어진 트리에 대한 설명으로 옳은 것은? (4.1점)



- ① 높이는 5이다.
 ② 포화이진트리(full binary tree)이다.
 ③ 단말 노드들은 1, 4, 11, 14, 23이다.
 ④ 편향이진트리(skewed binary tree)이다.
 ⑤ 노드 삽입은 완전이진트리(complete binary tree)를 유지하도록 삽입된다.

17. 다음과 같은 빈도로 문자가 사용되었을 때, 각 문자의 허프만코드로 옳지 않은 것은? (4.1점)

문자	b	e	g	j	l	o	s	u	z
빈도수	18	8	27	13	36	2	20	6	3

- ① b: 011 ② g: 00 ③ l: 10 ④ s: 111 ⑤ z: 11010

18. 다음 프로그램 실행 결과로 출력되는 *의 개수는? (3.9점)

```
int main(void) {
    int i = 0, j = 0;
    for (i = 0; i < 10; i++) {
        if (i % 3 == 0)
            continue;
        if (i > 8)
            break;
        for (j = 0; j < i; j++)
            printf("*");
    }
}
```

- ① 26 ② 27 ③ 28 ④ 29 ⑤ 30

19. 다음 프로그램을 실행시켰을 때 출력값은? (4점)

```
#include <stdio.h>
#include <string.h>
void main() {
    int array[256];
    char *str = "Boyer-Moore algorithm";
    int len = strlen(str);
    int i, j = 0;
    for (i = 0; i < 256; i++) array[i] = 0;
    while (j < len) {
        array[str[j]] = j;
        j = j + 1;
    }
    printf("%d", array['o']);
}
```

- ① 1 ② 2 ③ 4 ④ 15 ⑤ 21

20. 다음 프로그램을 식으로 표현한 것은? (4점)

```
#include <stdio.h>
void main() {
    int i, j, s, n;
    s = 0;
    scanf("%d", &n);
    for (i = 1; i <= n; i++)
        for (j = 1; j <= 2; j++)
            s = s + i;
    printf("%d", s);
}
```

- ① $\sum_{k=1}^n k$ ② $\sum_{k=1}^n 2k$ ③ $\sum_{k=1}^{2n} k$ ④ $\sum_{k=1}^n \sum_{m=1}^2 m$ ⑤ $\sum_{k=1}^n \sum_{m=1}^n m$

21. 다음은 삽입 정렬 프로그램이다. 밑줄 친 ㉠, ㉡에 들어갈 내용으로 옳은 것은?
 (단, 오름차순으로 정렬된다.) (4.1점)

```

void insertion_sort(int list[], int size) {
    int i, j, temp, a;
    for (i = 1; i < size; i++) {
        temp = list[i];
        j = i;
        while (_____ ㉠ _____)
            list[j + 1] = list[j];
        _____ ㉡ _____;
        for (a = 0; a < size; a++)
            printf("%d ", list[a]);
        printf("\n");
    }
}

int main(void) {
    int list[5] = { 5, 3, 1, 8, 7 };
    insertion_sort(list, 5);
}
    
```

- | | ㉠ | ㉡ |
|---|----------------------------|--------------------|
| ① | --j >= 0 && temp < list[j] | list[j] = temp |
| ② | j-- >= 0 && temp > list[j] | list[j - 1] = temp |
| ③ | --j >= 0 && temp > list[j] | list[j] = temp |
| ④ | j-- >= 0 && temp < list[j] | list[j + 1] = temp |
| ⑤ | --j >= 0 && temp < list[j] | list[j + 1] = temp |

22. 다음 프로그램을 실행시켰을 때 출력값은? (4.3점)

```
#include <stdio.h>
int A(int m, int n) {
    if (m == 0) return n+1;
    if (n == 0) return A(m-1,1);
    else return A(m-1, A(m,n-1));
}
void main() {
    printf("%d", A(2,3));
}
```

① 5

② 6

③ 7

④ 8

⑤ 9

23. 다음 프로그램을 실행시켰을 때 출력값은? (4.4점)

```
#include <stdio.h>
int rsa(int a, int b, int c) {
    int r;
    if (b == 1)
        return a % c;
    if (b%2 == 0) {
        r = rsa(a, b/2, c);
        return r*r % c;
    }
    else {
        r = rsa(a, b-1, c);
        return a*r % c;
    }
}
void main() {
    int ret;
    ret = rsa(120, 33, 37);
    printf("%d", ret);
}
```

① 10

② 12

③ 16

④ 33

⑤ 34

24. $n \times n$ 체스판에 n 개의 퀸(queen)을 배치하고자 한다. 퀸들은 같은 행이나, 같은 열이나, 같은 대각선상에 위치하지 않아야 한다. 이를 백트래킹으로 해결하고자 한다. 다음은 프로그램의 의사코드(pseudo code)일 때, promising 함수의 밑줄 친 ㉠에 들어갈 내용으로 옳은 것은? (4.5점)

```
// col[i] = i번째 행에서 퀸이 놓여있는 열
void queens(index i) {
    index j;
    if(promising(i))
        if(i == n)
            print col[1] through col[n];
        else
            for(j=1; j <= n; j++) {
                col[i+1] = j;
                queens(i+1);
            }
}
bool promising(index i) {
    index k = 1;
    bool switch = TRUE;
    while(k < i && switch) {
        if(_____ ㉠ _____)
            switch = FALSE;
        k++;
    }
    return switch;
}
```

- ① $col[i] == col[k] \ || \ |col[i]+col[k]| == |i+k|$
- ② $col[i] != col[k] \ \&\& \ |col[i]-col[k]| == |i-k|$
- ③ $col[i] != col[k] \ \ || \ |col[i]-col[k]| == |i-k|$
- ④ $col[i] == col[k] \ \ \&\& \ |col[i]-col[k]| == |i-k|$
- ⑤ $col[i] == col[k] \ \ || \ |col[i]-col[k]| == |i-k|$

25. 다음 입력에 따른 프로그램의 실행 결과는? (4.5점)

```

int n, v[101], g[101][101];
void f( int p ) {
    int i;
    v[p] = 1;
    printf("%c ", p + 64);
    for(i = 1; i <= n; i++) {
        if( ( v[i] == 0 ) && ( g[p][i] == 1 ) )
            f(i);
    }
}
int main() {
    int i, j;
    scanf("%d\n", &n);
    for(i = 1; i <= n; i++) {
        for(j = 1; j <= n; j++)
            scanf("%d", &g[i][j]);
    }
    f(1);
    return 0;
}

```

입력	<pre> 5 0 0 0 1 1 0 0 0 0 1 0 0 0 1 0 1 0 1 0 1 1 1 0 1 0 </pre>
----	--

- ① A B C D E
- ② A E D B E
- ③ A D C E B
- ④ A E B E C
- ⑤ A D E C B