

## ICP Final Exam

- 1 Let **a** and **b** be two **int** variables, and consider

```
if (a<b) { int z=a; a=b; b=z; } else ;
```

Explain why it is incorrect to rewrite this **if** statement as the following **switch** statement, and correct it.

```
switch (a<b) {
case true: int z=a; a=b; b=z; break;
case false:;
}
```

Problem	Score	Total
1~10	6% each	60%
11~12	8% each	16%
13	9%	9%
14	15%	15%

- 2 Show the output of the following code

```
int x;
void p(int a)
{
    static int b=a; int c=a; x++;
    printf("%d%d%d\n",x,b,c);
}
int main(void) { p(2); p(3); }
```

- 3 What is the type of each expression below?

- a) **p-a**                      given **int a[3], \*p=a+3;**  
b) **a+1**                      given **int \*a[2][3]**

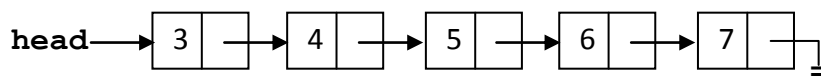
- 4 Given

```
int a[3][3]={1,2,3,4,5,6,7,8,9};
```

What is the value of each expression below?

- a) **\*(a+1)[1]**  
b) **a[1]-a[2]**

- 5 Given the linked list



where each node in the linked list is of the type

```
struct node { int datum; struct node* succ; };
```

- a) Write a piece of code to insert a node containing 2 before the node pointed to by **head**.  
b) Write a piece of code to delete the node pointed to by **head**.

- 6 a) The following incomplete function is meant to print out a  $3 \times n$  array **a**, for any  $n \geq 1$ . Fill in the blank that should contain the declaration of array **a**.

```
void print(_____, int n)
{
    for (int i=0; i<3; i++) {
        for (int j=0; j<n; j++) printf("%d ", (*a)[i][j]);
        printf("\n");
    }
}
```

- b) Given

```
int c[3][2]={1,2,3,4,5,6};
```

Write a piece of code to print out the  $3 \times 2$  array **c** by the function of part a).

- 7 Rewrite the following the loop using **if** and **goto** statements.

```
for (int i=1; i<=9; i++) print("%d", i);
```

- 8 Write the function

```
double randd(double x);
```

to generate a floating-point number in the range  $[0, x)$  uniformly at random.

- 9 Write the *recursive* function

```
int sum(int a[], int n);
```

to compute the sum of the **n** elements of the array **a**.

- 10 Write the function

```
int strcmp(const char* s, const char* t)
```

to compare two strings **s** and **t** so that **strcmp(s, t)** < 0, if **s** < **t**; = 0, if **s** = **t**; > 0, if **s** > **t**.

- 11 Given (8%)

```
int a[3]={0,1,2};
```

```
int* p=a+1;
```

For each expression below, show its value and draw a diagram showing the contents of the array **a** and the element pointed to by the pointer **p**.

- a) **\*p--**    b) **(\*p)--**    c) **\*--p**    d) **--\*p**

4
5
2
1
3

- 12 Consider sorting the array of integers into non-decreasing order:

- a) Show the contents of the array after making each pass over the data by **selection sort**, as given in lecture. Identify the sorted and unsorted subarrays. (4%)

12 (Cont'd)

- b) List all the comparisons between array elements that are done in the course of sorting the array by **bubble sort**, as given in lecture. (4%)

13 Let **exam.exe** be the executable of a C program whose function **main** is declared as

```
int main(int argc, char **argv);
```

Suppose the following command is entered within a command interpreter:

```
prompt> exam Snoopy Pluto SnoopyPluto
```

- a) What is the value of **argc**? (2%)
- b) Draw a picture showing the structure bound to **argv**, assuming that *string literals are shared*. (4%)
- c) Draw a picture showing the simplest structure that can be bound to a variable of type **char\*\***. (3%)

14 Consider the following simplified code for permutation generation discussed in lecture:

```
int a[3]={1,2,3};    // initialize the array
void perm(int i,int n)
{
    if (i==n) ;        // omit code for printing the array a[0..n]
    else {
        perm(i+1,n);
        for (int k=i+1;k<=n;k++) {
            int z=a[i]; a[i]=a[k]; a[k]=z;
            perm(i+1,n);
            z=a[i]; a[i]=a[k]; a[k]=z;           // line b
        }
    }
}
int main(void) {
    perm(0,2);
    for (int j=0;j<=2;j++) printf("%d",a[j]);    // line c
}
```

- a) Draw the recursion tree for the call **perm(0,2)**. (6%)  
Write down the contents of the array **a** alongside each node of the tree.
- b) Write a function **swap** to exchange the values of two **int** variables passing to it. (4%)  
Show how to replace the code in line b with a call to **swap**. (2%)
- c) Show the output of the code in line c. (3%)