Short-answered Questions: (6% each)

1

Write the following parameterized macros.

(a) `CHECK(x,y,n)` – Has the value 1 if both x and y fall between 0 and n – 1, inclusive.

(b) `MEDIAN(x,y,z)` – Finds the median of x, y, and z.

2. Declare an array in which the elements are either an integer or a string

3. How many bits are typically used for the size of a storage unit?

4.

```
union {
  double a;
  struct {
    char b[4];
    double c;
    int d;
  } e;
  char f[4];
}
```

If `char` values occupy one byte, `int` values occupy four bytes, and `double` values occupy eight bytes, how much space will a C compiler allocate for u? (Assume that the compiler leaves no "holes" between members.)

5

Suppose that `f` and `p` are declared as follows:

```
struct {
  union {
    char a, b;
    int c;
  } d;
  int e[5];
} f, *p = &f;
```

Which of the following statements are legal?

(a) `p->b = ' ';`

(b) `p->e[3] = 10;`

(c) `(*p).d.a = '*';`

(d) `p->d->c = 20;`

6.

The following loop is supposed to delete all nodes from a linked list and release the memory that they occupy. Unfortunately, the loop is incorrect. Explain what's wrong with it and show how to fix the bug.

```
for (p = first; p != NULL; p = p->next)
  free(p);
```

7

Write a function that, when given a string as its argument, searches the following array of structures for a matching command name, then calls the function associated with that name.

```
struct {
  char *cmd_name;
  void (*cmd_pointer)(void);
} file_cmd[] =
 {{"new",        new_cmd},
  {"open",       open_cmd},
  {"close",      close_cmd},
  {"close all", close_all_cmd},
  {"save",       save_cmd},
  {"save as",    save_as_cmd},
  {"save all",   save_all_cmd},
  {"print",      print_cmd},
  {"exit",       exit_cmd}
 };
```

8. What are the purposes of using a bit mask?

9. Explain how you would compare two structures for equality.

10. What objective does each of the following macros achieve?

#define unknown1(w, n, k) (((unsigned int) (w) & (~(~0 << (k)) << (n))) >> (n))

#define unknown2(w, n, k, v) (((unsigned int) (w) & ~(~(~0 << (k)) << (n))) | ((unsigned int) (v) << (n)))

Programming Problems: (20% each)

1. Write an iterative function that reverses the links of a linked list ***in place***, that is, you are not allowed to call malloc() or declare an array in the function. The prototype of the function is as follows:

Node * reverse(Node *head)

in which ***Node*** is a user-defined structure type that contains a pointer field ***link*** pointing to the next node on the list and ***head*** is a pointer that points to the start of the linked list. The function returns a pointer to the node that is the head of the reversed linked list.

2. Write a generalized version of binary search that searches an array of integers, strings, or structures that specify a point in 3D space. The prototype of the function is as follows:

void * bsearch(void * list, size_t N, size_t size, void *target, int (*compare)(const void *, const void *))

in which **list** is the array to be searched, ***N*** is the number of elements in list, ***size*** is the space occupied by each element of list, ***target*** is a pointer to the object of search, and ***compare*** is a function pointer to a function that performs the necessary comparison during search and returns 1 if its first argument is greater than the second argument, 0 if the two arguments are equal, and -1 if the first argument is less than the second argument. The function returns a pointer to an element in list if search is successful, or NULL if search fails.