

**Final**

MAY 10, 5:00–7:30 PM

There are 8 problems each worth five points for a total of 40 points. Show all your work, partial credit will be awarded. Space is provided on the test for your work; if you use a blue book for additional workspace, sign it and return it with the test. No notes, no collaboration.

Name: \_\_\_\_\_

Problem	Credit
1	
2	
3	
4	
5	
6	
7	
8	
Total	

1. Write a declaration statement declaring a variable of name and type given. For instance, if the question asks for an integer variable `i`, the answer is `int i;`
  - (a) A pointer to integer type with name `ip`.
  - (b) A pointer to an array of strings, named `argv`.
  - (c) An array of of 10 strings, name `sa`.
  - (d) An array of 10 pointers to integers, named `ipa`.
  - (e) A variable `fp`, a pointer to a function taking an integer argument and returning a pointer to a character.
  - (f) A pointer to an array of pointers to functions, each taking a pointer to a structure with structure tag `s` as an argument, and returning void. The length of the array is unknown (arbitrary). Call this variable `fpa`.

2. Write a declaration statement declaring a structure of tag and type given. For instance, if the question asks for a structure `s` with one integer field `i` the answer is `struct s { int i ; };`
  - (a) A structure with two fields, `a` and `a_len`, the first of type array of `SIZE` integers, the second of type integer, initialized to the value `SIZE`.
  - (b) A structure with tag `s` with two fields, `val` of type pointer to integer, and `next` of type pointer to this structure type. Declare no variables.
  - (c) A structure `s_out` containing two structures `s_in_a` and `s_in_b`. Structure `s_in_a` contains a single field named `a` of type integer; structure `s_in_b` contains a single field named `b` of type integer. (Do this with a single declaration.)

3. Write a C function taking an array of integers and the length of the array, and returning the (integer) sum of all elements in the array. If the length of the array is unreasonable, or for any other error, return 0. You only need to write the function. Do not write `main`!

4. Write a C function `strcatMalloc` taking two strings and returning a newly created string which is the concatenation of the argument strings. The function calls `malloc` to get the needed memory for the new string.

5. True or false:

- (a) Sockets are for network communication.
- (b) Sockets can be used only for communication between two different computers.
- (c) A socket can be used for reading only, or for writing only, not both.
- (d) A datagram socket, using UDP, guarantees delivery of the data.
- (e) The server prepares for communication by doing a passive listen on the TCP endpoint.
- (f) When the call `accept` returns, it has created a new socket for communication.
- (g) All operating systems treat the low number ports (lower than port number 1024) as privileged.
- (h) The Internet was created by the Department of Defense in collaboration with the Massachusetts Institute of Technology (MIT).

6. Why will the following use of signals not work reliably,

```
f = fopen( "myfile", "w" ) ;
if ( (pid = fork()) == 0 ) {
    fprintf( f, "%d", getuid() ) ;
    exit(0) ;
}
kill( pid, SIGKILL ) ;
fclose(f) ;
```

7. What does this program do?

```
#include<signal.h>

char * s, * t ;

void f(int j){ printf("%d\n", s-t) ; exit(0) ; }

void g(void) { if (*s) s++ ; else kill(getpid(),SIGINT) ; g() ; }

int main(int argc, char * argv[]) {
    signal(SIGINT, f) ; s = t = argv[0] ; g() ;
}
```



8. Consider the string linked list data structure and utility routines:

```
struct sll { char * s ; struct sll * next ; } ;

struct sll * sllCreate( char * s ) {
    struct sll * sll = (struct sll *) malloc(sizeof(struct sll)) ;
    sll->s = strdup( s ) ;
    sll->next = 0L ;
    return sll ;
}

struct sll * sllCopy( struct sll * sll_src ) {
    struct sll * sll = (struct sll *) malloc(sizeof(struct sll)) ;
    sll->s = sll_src->s ;
    sll->next = sll_src->next ;
}

void sllDelete( struct sll * sll ) {
    free(sll->s) ;
    free( sll ) ;
}
```

What problem will the following code encounter? How can this be prevented?

```
int main(int argc, char * argv[]) {
    struct sll * s1, * s2 ;
    s1 = sllCreate("hi") ;
    s2 = sllCopy(s1) ;
    sllDelete(s1) ;
    printf("%s,\n", s2->s ) ;
}
```