

4/3/0

1. What is missing from the array declaration below for **string**? Explain why it is needed.

```
char string[] = {'B','e','a','t',' ','A','r','m','y'};
```

6/4/2/0 2. For **string** declaration:

```
char string[] = "Crash Airforce"
```

What is the length of the string?

Explain

How much memory (in bytes) is required to store the string?

3. What is the output of the code snippet below, and explain:

```
char string[] = {'B','e','a','t',' ','A','r','m','y','\0',  
                'C','r','a','s','h',' ','A','i','r','f','o','r','c','e','\0'};  
printf("%s",string);
```

4/2/0

6/4/2/0 4. Complete the program below to copy **str1** to **str2**.

```
int main(){  
  
    char str1[]="I love IC221!";  
    char str2[  ]; //declare str2  
    int i;  
  
    //copy str1 to str2  
    for(i=0 ;  ; i++){  
  
  
    }  
}
```

5. Use the Unix manual pages to look and describe the following **string.h** library functions and its arguments:

4/2/0

a) strcpy ()

4/2/0

b) strncpy ()

4/2/0

c) strchr ()

4/2/0

d) strrchr ()

4/2/0

e) strcat ()

4/2/0

f) strncat ()

4/2/0

g) strfry ()

2/1/0

6. For the string library functions contain an "n" as in **strncpy()**, explain why specify the length of the destination string is important?

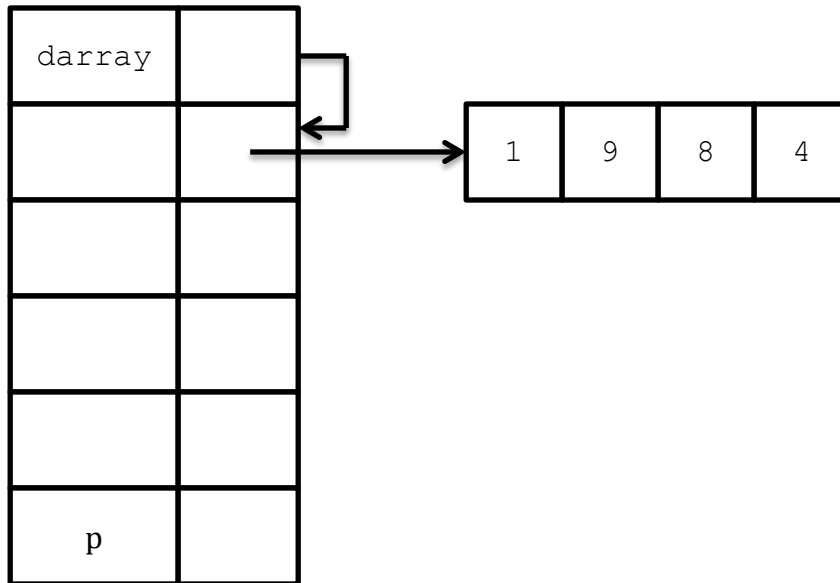
4/2/0 7. What is the output of the following program?

```
int main(){
    int darray[][4] = {{1, 9, 8, 4},
                       {1, 8, 9, 4},
                       {2, 0, 1, 7},
                       {3, 4, 5, 8}};

    int * p = &(darray[1]);

    printf("%d",p[2]);
}
```

7/5/3/0 8. For the above program, complete the stack diagram. Fill in missing values and data segments.



5/3/0 9. Convert the following double array dereference with [] to one without []

```
darray[3][0] 
```

8/6/3/0 10. Write a for loop to iterate over the **darray** to print a 4x4 grid of numbers, much like it is presented in question 7 above.

4/2/0

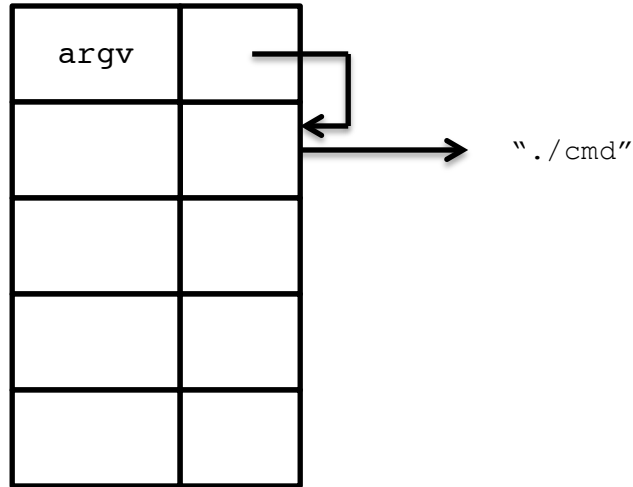
11. Describe the following type declaration with respect to pointers and arrays.

```
char * strings[]
```

7/5/2/0

12. Complete the argv array below for call in the program with following command line arguments:

./cmd go navy



8/5/3/0

13. Explain why the following for loop iteration over the **argv** array will terminate:

```
int main(int argc, char * argv[]){
    char ** curarg;
    int i;

    i=0;
    for( curarg=argv;
        *curarg != NULL;
        curarg++, ){
        printf("argv[%d] = %s\n", i, *curarg);
        i++;
    }
}
```

3/1/0

14. Using atoi() to convert the string to a number.

```
char five[] = "5"
int i; //set to five
```

15. Using sscanf() to convert the string to a number

3/1/0

```
char five[] = "5"
int i; //set to five
```