61.5

CSC 111 Fall 2011
Midterm 2

- Turn in your completed midterm at the front of the class; leave through the front door on your left.
- Do not leave before 10:45 am.

1. Consider the following syntactically correct C declarations and assignments. **[8]**

```
int a;
int d;
int *b;
int *c;
int** e;
a = 17;   b = &a;
c = b;    d = 19;
e = &c;
```

What are the values of the following expressions?

(&a == c)	True
(b == &d)	False
(*c == 17)	True
(**e == *b)	True
(d == a)	False
(*b == 19)	False
(a == b)	Invalid
(&a == e)	True

1

2. Consider the following declarations: **[6]**

```
typedef struct {
        int year;
        int month;
        int day;
} Date;
Date dob;
Date *d = &dob;
```

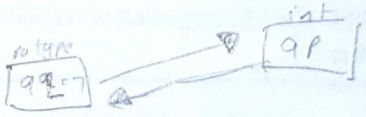
Using variable **d** initialize **dob** with the following birthday **July 1, 1867.**

6

```
d->year = 1867;
d->month = 7;
d->day = 1;
```

3. Consider the following declarations: **[4]**

```
int* ap;
void* aq;
```



How do you assign **aq** to **ap** properly using a cast?

8

```
void* aq = 7;
*aq = &ap;
*ap = &aq;
```


b

4. Consider the following syntactically correct C declarations and assignments. **[6]**

```
int a;
int *b
int **c;
a = 17;
b = &a;
c = &b;
```

Using one **printf()** statement output the **address** of variable **b** and the **address** of variable a.

6

```
printf(" address of b: %d \n address of a: %d \n", c, b);
```

5. In the C programming language, how do you refer to a file when you read, write or close a file? **[4]**

- (X) FILE*
- ( ) fopen
- ( ) printf
- ( ) fgetc

4

6. What will happen if you execute the following C program? **[4]**

```
#include<stdio.h>
#include<stdlib.h>
int main(void) {
   FILE *fp1,*fp2;
   fp1 = fopen("day.txt", "r");
   fp2 = fopen("night.txt", "r");
   fclose(fp1, fp2);
   return EXIT_SUCCESS;
} /*main*/
```

4

- (X) Compile-time error
- ( ) It will open two files in read mode
- ( ) It will check whether the two files "day.txt" and "night.txt" exist
- ( ) It will open two files in read mode and then close them.

14

7. What is the console output of the following syntactically correct C program? **[6]**

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
int main(void){
        char str[] = "Once upon a time there was a polar bear";
        FILE *ifp;
        FILE *ofp;

        ofp = fopen("vic.txt", "w");
        fputs("I live in Victoria!\n", ofp);
        fclose(ofp);
        ifp = fopen("vic.txt", "r");
        while(fgets(str, strlen(str), ifp) != NULL)  printf("%s", str);
        fclose(ifp);
        return EXIT_SUCCESS;
} /* main */
```

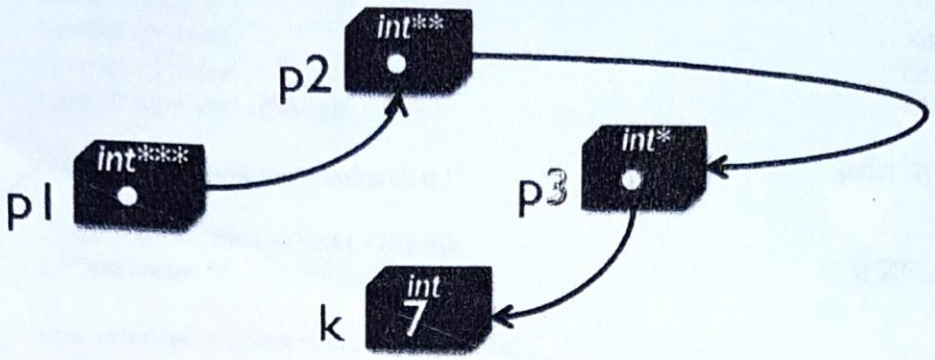
- ○ Once upon a time there was a polar bear
- ○ vic.txt
- ⊗ I live in Victoria!
- ○ None of the above

8. What is the output of the following syntactically correct C program? **[6]**

```
#include <stdio.h>
int main(void) {
int k = 5;
while (k < 12) {
    printf("%d ",  k%7);
    k = k + 1;
} /* while */
printf("\n");
return 0;
} /* main */
```

Output: 5 6 7 8 9 10 11

9. Realize the following memory configuration using C variable declarations and pointer assignments. Write three assignments using pointers p1, p2, and p3 to store 7 in variable k? **[8]**




```
int***** p1 = 7;
***** p1 = & p2;
**** p2 = & p3;
*** p3 = & k;
```

10. Which of the following functions is used to read file on a char by char basis? **[4]**

- ○ putc(int, FILE*)
- ○ char* fgets(char*, int, FILE *)
- ○ fscanf("format", args)
- ○ int getc(FILE *)

11. Which of the following is **not** true? **[4]**

- ○ Each component of a union is assigned the same chunk of storage space
- ○ The syntax for unions is basically the same as for structures
- ○ Each component of a union is of same type
- ○ Unions are also called variant records

printf()

12. Insert a syntactically correct **print()** statement into the following C code—where the box is—to output the **second to last** character of the string **s.str** using the length of the string. **[6]**

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define MAX_SIZE  (100)
typedef struct {
   int length;
   char str[MAX_SIZE ];
}  StringDesc;

int main(void) {
   StringDesc s;
   strcpy(s.str, "Melanie Amaro of X FACTOR");
   s.length = strlen(s.str);
```

4

printf("%c\n", s.str[length-2]);

%c 2

```
   return EXIT_SUCCESS;
} /* main */
```

13. What is the output of the following syntactically correct C program? **[6]**

X

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
int main(void) {
      char song2[] = "Listen Beyonce";
      /* strcmp()returns an int greater than, equal to, or less than 0, if the first  string */
      /*  is greater than, equal to, or less than the second string respectively. */
      if (strcmp(song2,  "Beyonce Listen") == 0) printf("Beyonce\n");
      else printf("Amaro\n");
      return EXIT_SUCCESS;
} /* main */
```

0

Output: Beyonce

4

14. What is the effect of the following initialization? **[8]**

```
#include <stdio.h>
#define  vSize  3
typedef int Item;
typedef int index;
typedef Item Vector[vSize];

void initVector(Vector V, index size) {
   index k;
   for (k=0; k<size; k++) V[k] = (Item)k;
} /* initVector */

void printVector(const Vector V, index size) {
   index k;
   for (k=0; k<size; k++) printf("%d  ", V[k]);
   printf("\n");
} /* printVector */

int main(void) {
   Vector Vec;
   initVector(Vec, vSize);
   printVector(Vec, vSize);
   return 0;
} / * main */
```

- ◯ 1 2 3
- ⊗ 0 1 2
- ◯ 1 1 1
- ◯ 0 0 0

15. How do you swap the values in variables **a** and **b** using the routine swap? **[4]**

```
void swap(int* x, int* y) {
   int tmp = *x; *x = *y; *y = tmp;
} /* swamp */
```

- ◯ int a = 3, b = 17; swap (3, 17);
- ◯ int a = 3, b = 17; swap(a, b);
- ⊗ int a = 3, b = 17; swap(&a, &b);
- ◯ int a = 17, b = 3;

16. What is the output of the following syntactically correct C program? **[8]**

```
#include <stdio.h>
typedef float Vector[3];
void func1(Vector a, int len) {
        int k;    float first = a[0];
        for (k=0; k<len-1; k++) {   a[k] = a[k+1];   printf(" %.1f", a[k]);  }
        a[len-1] = first; printf(" %.1f", a[len-1]);
} /*func1*/
int main(void) {
        Vector vec;
         vec[0] = 1.1;    vec[1] = 5.5;     vec[2] = 4.4;
        func1(vec, 3);
        return 0;
} /* main */
```

5.5 4.4 1.1

8

- ○ 1.1 4.4 5.5
- ○ 1.1 5.5 4.4
- ⊗ 5.5 4.4 1.1
- ○ 5.5 4.4

17. Write a syntactically correct C program to open two text files: an input file called **Beatles.txt** and an output file called **RollingStones.txt**. Copy the contents of file **Beatles.txt** to **RollingStones.txt**. Then close the files. **[8]**

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
int main(void){
  FILE *ifp;
  FILE *ofp;
  ifp = fopen("Beatles.txt","r");
  ofp = fopen("RollingStones.txt","w");
  fputs(ifp, ofp);
  fclose(ifp);
  fclose(ofp);
  return EXIT_SUCCESS;
}
```

4.5

1 1 0 1

1.5

12.5