Questions

1. [6 marks] What is the **console output** of the following syntactically correct C program?

```c
#include <stdio.h>
#include <stdlib.h>
#define FNAM ("UVic.html")
#define SIZE (100)
int main(void){
    FILE *ifp;
    FILE *ofp;
    char str[SIZE];
    ofp = fopen(FNAM, "w");
    printf("I can generate web pages!\n");
    fprintf(ofp, "<html><head><title>CSC111");
    fprintf(ofp, "</title></head>\n");
    fprintf(ofp, "<body><h1>Victoria is beautiful!");
    fprintf(ofp, "</h1></body></html>\n");
    fclose(ofp);
    ifp = fopen(FNAM, "r");
    while(fgets(str, SIZE, ifp) != NULL)
        printf("%s", str);
    fclose(ifp);
    return EXIT_SUCCESS;
}/*main*/
```

I can generate web pages!
<html><head><title>CSC111

3.

2. [6 marks] What is the console output of the following syntactically correct C program?

```c
#include <stdio.h>
#include <stdlib.h>
#define LEN (4)
int main() {
    double a[LEN]; //LEN is a constant
    double d = 9.99;
    for (int k=LEN-1; k>=0; k--){
        a[k] = d; d = d - 1.11;
    }/*for*/
    a[0] = 9.99; a[1] = 2.77;
    a[2] = 0.07; a[3] = 7.87;
    double b[LEN]={2.81, 0.23, 0.89, 7.87};
    double* dp = b;
    *dp = 2.81; dp++; *dp= 3.14; dp--; *dp= 4.66;
    for (int k=0; k<LEN; k++)
        printf("%.2f %.2f\n", a[k], b[k]);
    return EXIT_SUCCESS;
}/*main*/
```

9.99 4.66 2.77 4.66 0.07 4.66 9.11

4.66

3

9.99 2.87
2.77 0.23
0.07 0.89
7.87 7.87

4.5

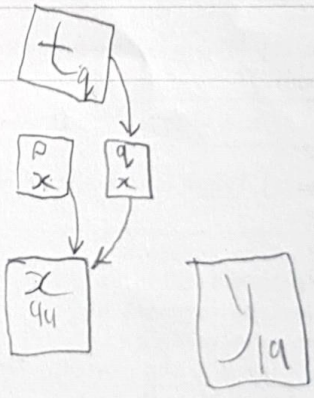
3. [8 marks] Write a syntactically correct C function to add an integer value `val` to all the elements of a one-dimensional array `A` with `len` elements. For example: **A[3 6 5 2 1] + 3** is **A[6 9 8 5 4]**

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  void addValueArray(int A[],  int len, int val) {
4    // Your code goes here
5    for(int k=0; k<len; k++){
6      A[k] = A[k] + val;
7      }/*for*/
8
9
10
11
12
13 }/*addValueArray*/
```

8

4. [8 marks] Consider the following syntactically correct C declarations and assignments:

```
1  int x; int y; int *p; int *q; int** t;
2  x = 44; p = &x; q = p; y = 19; t = &q;
```



What are the values of the following expressions?

(a) True False (**t== *p)
(b) True False (y == x)
(c) True False (*q == 44)
(d) True False (&x == p)
(e) True False (&x == *t)
(f) True False (p == &y)
(g) True False (*q == *p)
(h) True False (*q == 17)

8

5. [8 marks] Write a complete syntactically correct C program (not only `printf` calls) that uses the formats %s and %d to output two strings and six integers vertically aligned as follows:

```
1  @#$%DEJA VU%$#@!*
2   1234567 2345678
3     34567 45678
4       567 678
```

20.5

```
#include<stdlib.h>
#include<stdio.h>
void main(){
  char A='@#$% DEJA';
  char B='VU%$#@!*';
  int c=1234567;
  int d=23456 78;
  int e=34567;
  int f=45678;
int g=567;
int h=678;
printf("%s %s",A,B);
printf("%8d %d",c,d);
Printf("%8d %d",e,f);
printf("%8d %d",g,h);
return EXIT_SUCCESS;
}/*main*/
```

6

6. [8 marks] Consider the C programming language, for each of the statements below mark the appropriate answer **True** or **False**.

(a)	True	False	Passing pointers as parameters to a function allows to get values back from a function
(b)	True	False	An expression is true if its value is zero
(c)	True	False	The call `fopen("Pumpkin.txt", "w");` will return `NULL` if the file `"Pumpkin.txt"` exists
(d)	True	False	The type of a pointer variable may contain at most three * characters
(e)	True	False	Parameterization and abstraction are powerful tools for engineers and scientists
(f)	True	False	All the components of an array are of the same type
(g)	True	False	An HTML/SVG file is a text file and can be viewed in a web browser
(h)	True	False	A for loop may execute its loop body zero times

7

7. [4 marks] Which of the following expressions generates a pseudo random integer in the range 17 to 35 (inclusive)?

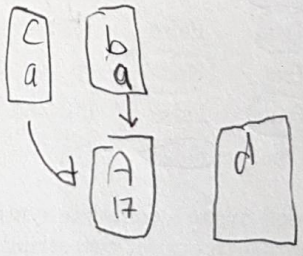
A. `rand()%19 + 17` B. `rand()%(35 - 16)+ 18` C. `rand()%35 + 17` D. `rand()%18 + 18`

35 − 17 = 18

4

8. [4 marks] What is the output of the following syntactically correct C program?

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <stdbool.h>
4  int main(void) {
5      int a = 17;  int d;  int* b = &a;  int* c = b;
6      bool b1 = (b == &d); false
7      bool b2 = (b == c); true
8      bool b3 = (&a == c); true
9      bool b4 = (*b == 19); false
10     if (b1) printf("true "); else printf("false ");
11     if (b2) printf("true "); else printf("false ");
12     if (b3) printf("true "); else printf("false ");
13     if (b4) printf("true "); else printf("false ");
14     printf("\n");
15     return EXIT_SUCCESS;
16 }/*main*/
```



4

false true true false

21

9. [4 marks] Which of the following `printf()` format specifications is used to print an address in hexadecimal number in the programming language C?

A. `%8s` B. `%p` C. `%-4d` D. `%x`

10. [4 marks] Which sequence of operators has the correct precedence order—*from highest to lowest*—in the programming language C?

A. [] ! %= <= B. ++ % || []
C. () < && += D. % + == []

11. [6 marks] What is the output of the following syntactically correct C program?

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  int f1(int x, int y, int z){
4      return x + y - z;
5  }/*f1*/
6  void f2(int* a, int* b){
7      int t = *a + *b;
8      *a = t + 6;
9      *b = t - 7;
10 }/*f2*/
11 void f3(int* p, int q){
12     int u = *p - q;
13     *p = u * 3;
14 }/*f3*/
15 void printInts(int a, int b, int c) {
16     printf("a = %2d, b = %2d, c = %2d\n", a, b, c);
17 }/*printInts*/
18 int main(void) {
19     int a = 9, b = 9, c = 9;
20     a = f1(a, b, c);     printInts(a, b, c);
21     f2(&b, &c);          printInts(a, b, c);
22     f3(&a, c);           printInts(a, b, c);
23     return EXIT_SUCCESS;
24 }/*main*/
```

a = 9, b = 9, c = 9
a = 9, b = 24, c = 11
a = -6, b = 24, c = 11

14

12. [6 marks] Consider the partially written syntactically correct C code below.

```
#include <stdio.h>
#include <stdlib.h>
void rotateLeft(int* x, int* y, int* z){
  // Your code goes here
  int temp = *x;
  *x = *y;
  *y = *z;
  *z = temp;

}/*rotateLeft*/

int main(void) {
    int a = 111, b = 116, c = 106;
    printf("%d %d %d\n", a, b, c);
    rotateLeft(&a, &b, &c);
    printf("%d %d %d\n", a, b, c);
    rotateLeft(&a, &b, &c);
    printf("%d %d %d\n", a, b, c);
    return EXIT_SUCCESS;
}/*main*/
```

6

In the space **above**, complete the definition of the `rotateLeft()` function such that the entire program is syntactically correct and produces the following output:

```
111 116 106
116 106 111
106 111 116
```

13. [8 marks] Write a syntactically correct C function to reverse the elements of a one-dimensional array `A` with `len` elements. For example: **A[3 6 9 2 7]** reversed is **A[7 2 9 6 3]**

0 1 2 3 4 5

```
#include <stdio.h>
#include <stdlib.h>
void reverseArray(int A[],  int len) {
  // Your code goes here
  int temp[len];
  for (int k=0; k<len; k++) temp[k] = A[k];
  for (int n=0; n<len; n++) {
  if(n<((len-1)/2)) A[n] = temp[((len-1)-n)];
  else if(n==((len-1)/2)) A[n] = temp[n];
  else if(n>((len-1)/2)) A[n] = temp[((len-1)-n)];
  }/*for*/


}/*reverseArray*/
```

8

14