Fundamentals of Computing and Programming

End Term Examination. 70 marks, Time Limit 3 hrs

December 8, 2022

1. What is printed in each case of the given code snippet: [2x5=10]

(a)
```
int n=0;
n?printf("A\n"):printf("B\n");
n++;
n?printf("A\n"):printf("B\n");
```

(b)
```
int g=100;
void f(){
   int g=200;
}
void h(){
    printf("%d\n",g);
}
void main(){
   g=300;
   f();
   printf("%d\n",g);
   h();
}
```

(c)
```
int i=15;
while(i>1)
   printf("%f ",i=i/2);
```

(d)
```
int i=0b00001100;
while(i!=0) {
     printf("1 ");
     i=i>>1;
}
```

(e)
```
int *p;
int a[10]={1,2,3,4,5,6,7,8,9,10};
p=a+5;
printf("%d  %d",p[0],p[1]);
```

2. Answer briefly each of these: [2x5=10]

(a) Mention two ways in which a local variable of a function is different from a global variable (apart from saying where they are declared).

(b) If we inspect the output of compilation (eg "`cat a.out`", where `a.out` is an executable) the output is not comprehensible. Why?

(c) Which has a larger size and why: a **union** with three fields **or** a **structure** which exactly the same three fields.

(d) A character variable `ch` has some character in it. How do we use printf to print the ASCII value of that character?

(e) Explain in words why the following assignment is wrong:

```
int name[20];
name="abcdef";
```

3. [2+2+4+2=10]

(a) **Declare a structure** called `struct car` with fields: (i) `owner` which is a character string with atmost 20 characters (ii) `year` which is an integer (iii) `model` which is a string of atmost 20 characters.

(b) **Declare the prototype** of the function called `compare`. It takes **two** `struct car` parameter values (*not pointers*) and returns an integer.

(c) Write out the above `compare` function. It compares the two structure parameter values field by field and returns 1 if they are identical values and returns 0 otherwise.

(d) Study the function below, say what is wrong and rectify it so that it works correctly:

```
// This function returns a pointer to a string
// which contains the name of the owner
char * get_owner(struct car c){
   char name[25];
   strcpy(name,c.owner);
   return(name);
}
```

4. For this question assume that `struct car` of the previous question has been declared correctly. [2+5+3=10]

(a) Declare a variable which is an array of at most 100 such `struct car`.

(b) Write out a function with this prototype: `void insertion_sort(struct car a[], int n);` It takes an array of n car structures and sorts them using insertion sort. They are to be sorted by the `year` field.

(c) Using the big-O notation and mention the complexity of your code. In particular mention the number of moves and number of compares in your code in terms of the parameter `n`. Say how you arrived at your analysis.

5. Consider this declaration: [2+4+4=10]

```
struct node {
    int v;
    struct node *next;
};
```

(a) Draw a linked list of nodes with the values 3 2 1 4 in that order showing the two fields for all the nodes. Use arrows to show the pointer fields. Clearly mark the head of the linked list.

(b) Write a function to search a node in the linked list. It returns 1 if the node exists and 0 if it does not. Its parameters are the pointer to the head of the linked list and an integer value that needs to be searched for.

(c) Write a function to delete a node from the linked list. It returns a pointer to the head of the modified linked list. It's parameters are the pointer to the head of a linked list and an integer value that needs to be searched. The corresponding node is deleted if it is in the list. Otherwise the list is left unchanged. You may assume no repeated values exist in the linked list.

6. (a) [6+2+2=10]
Array `a[]` has `n` integers sorted in increasing order. Array `b[]` has `m` integers sorted in increasing order. Array `c[]` has space for `n+m` integers. Write the function: `void merge(int a[], int n, int b[], int m, int c[]);` It will merge the contents of `a[]` and `b[]` into `c[]`. Use the fact that `a[]` and `b[]` are sorted to created the sorted and merged array `c[]` when the function completes.

(b) In terms of `n` and `m` what is the complexity of this code (use the big-O notation)?

(c) One way to create `c[]` is to copy all elements of `a[]` and `b[]` to `c[]` one after the other, and then simply use bubblesort to sort `c[]`. What would be the complexity of this way to create c[] in terms of `n` and `m`. Explain.

7. Consider the following definition: [5+5=10]

$$C(n,r) = \begin{cases} 0, & n < r \\ 1, & n = r \\ 1, & n > r,\ r = 0 \\ C(n-1,r-1) + C(n-1,r) & n > r > 0 \end{cases}$$

Here $C(n, r)$ represents the number of ways to choose r objects from among n objects, with non-negative integers n and r.

(a) Use the equations above to write a recursive function called `compute_r()` which takes n and r as parameters and returns the value of $C(n, r)$.

(b) Assume a *square* matrix C to hold these values. Let the matrix be defined with 10 rows and 10 columns. We wish that $C[n][r]$ has the value of $C(n, r)$. Write a function `compute_m()` to receive as *parameters* the uninitialized matrix C and the value m. It compute all the entries for the matrix C from $C[0][0]$ to $C[m][m]$. without using recursion. Hint: Simply notice that elements in the n^{th} are computed by entries from the $n - 1^{th}$ row. Compute entries of C row by row.