IST-OPS Final Exam — 7th December 2023

Name:

- First, write your name in the box above. Then, have a quick read through all 6 questions.
- In the end, you will write up your answers on this paper.
  - But please make a draft elsewhere first. Only hand in something readable. Really.
- This is an open-book open-laptop exam: you may work on scrap paper and/or on your screen.

Question 1 Consider a directory hierarchy where running `ls -R` produces the output shown below on the left. In the rectangle on the right, give a sequence of `cd`, `rm`, `rmdir` commands that deletes all these files and directories.
Your answer should satisfy the following constraints:

- no error message i.e. all commands run successfully
- no "recursive `rm`" i.e. no `rm -r`
- no slash character i.e. no `/` in your commands

```
$ ls -R .
.:
pink  white

./pink:
brown

./white:
blue  orange

./white/blue:

./white/orange:
green
```

Question 2 Implement the function below so that it computes the sum of all the elements of an array "`t`" of length "`n`". You may not use the indexing operator "`[]`".

```
int sum(int *t, unsigned int n)
{
   int result = 0;









   return result;
}
```

Question 3 Use `printf()` to display number -1 in hexadecimal notation. What does it show on screen? Give a one sentence explanation of why this is the case.

your code:

output:

explanation:

Question 4 In the list below, which system call(s) return control to the calling code when successful, and which don't? Please circle "Y" for "this function does return" and circle "N" for "this function does not return".

Y	N	exec
Y	N	exit
Y	N	fork
Y	N	getpid
Y	N	sleep
Y	N	wait

Question 5 Consider the program on the right.

How many values are printed in total?

What is printed by the parent process?

What is printed by the first child process?

```
main()
{
  int i=0;
  if( fork() != 0)
  {
     i=i+1;
  }
  fork();
  i=i+2;
  printf("i=%d\n",i);
}
```

Question 6 Write a `firstdiff.c` prograw which reads two text files line-by-line and prints the first line where they differ. Your program should also show the relevant line number, as illustrated below.

Use `fgets()` from `<stdio.h>` to read text (you may assume a fixed maximum line length, e.g. 100 characters) and use `strcmp()` from `<string.h>` to compare lines.

```
$ echo -e 'Alpha\nBeta\nGamma\nDelta\nEpsilon' > A.txt
$ echo -e 'Alpha\nBeta\nUltraviolet\nDelta\nEpsilon' > B.txt
$ ./firstdiff A.txt B.txt
A.txt:3: Gamma
B.txt:3: Ultraviolet
$ echo Alpha > C.txt
$ ./firstdiff B.txt C.txt
B.txt:2: Beta
C.txt:2:
```

`firstdiff.c`