1. [3] Consider the C program below. Assume that all functions return normally.

```c
main() {
    if (fork() == 0) {
        if (fork() == 0) {
            printf("3");
        } else {
            pid_t pid; int status;
            if ((pid = wait(&status)) > 0) {
                printf("4");
            }
        }
    } else {
        if (fork() == 0) {
            printf("1");
            exit(0);
        }
        printf("2");
    }
    printf("0");
    return 0;
}
```

Out of the 5 outputs listed below, circle only the valid outputs (can be one or more or none) of this program. Assume that all processes run to normal completion.

A. 2030401  B. 1234000  C. 2300140  D. 2034012  E. 3200410
2. Consider the C program below. Assume that all functions return normally.

```c
int main () {
    if (fork() == 0) {
        if (fork() == 0) {
            printf("3");
        } else {
            pid_t pid; int status;
            if ((pid = wait(&status)) > 0) {
                printf("4");
            }
        }
    }
    printf("2");
    exit(0);
} else {
    printf("0");
    return 0;
}
```

What are the possible output(s) of that program?
3. Consider a computer system that has a cache with 256 blocks. Each block can store 16 bytes. What will be the value stored in the TAG field of the cache block that holds the memory block containing the address 0x3CFBCF (Note: The address provided tells you that the length of an address here is 24 bits)

(i) [3] if it is a direct-mapped cache

(ii) [3] if it is a 16-way set-associative cache

(iii) [3] if it is fully associative


```c
int val = 10;

void handler(sig)
{
    val += 7;
    return;
}

int main()
{
    int pid;
    signal(SIGCHLD, handler);
    if ((pid = fork()) == 0) {
        val -= 3;
        exit(0);
    }
    waitpid(pid, NULL, 0);
    printf("val = %d\n", val);
    exit(0);
}
```