Enrollment no	OS Practicals
U18CO028	You are given an integer n and an array of size n, where array[i] represents the ith dimension of an integer array A of n dimensions in c program. Find the number of page faults for a page size of 256 while iterating over the entire array A.

U18CO029	Consider Peterson’s algorithm for mutual exclusion between two concurrent process ‘i’ and ‘j’. The program executed by process ‘i’ is shown below.
	repeat
	flag[i] = true;
	turn=j
	while(--------p-----) do no-op;
	{
	Critical section;
	}
	flag[i] = false;
	Perform other non-critical section;
	until false;
	For the program to guarantee mutual exclusion, the predicate p in the while loop should be?

U18CO030	Assume that a finite number of resources of a single resource type must be managed. Processes may ask for a number of these resources and once finished-will return them. As an example, many commercial software packages provide a given number of licenses, indicating the number of applications that may run concurrently. When the application is started, the license count is decremented. When the application is terminated, the license count is incremented. If all licenses are in use, requests to start the application are denied. Such requests will only be granted when an existing license holder terminates the application and a license is returned.
	The following program segment is used to manage a finite number of instances of an available resource. The maximum number of resources and the number of available resources are declared as follows:
	#define MAX_RESOURCES 5
	int available_resources = MAX_RESOURCES;
	When a process wishes to obtain a number of resources, it invokes the decrease_count () function:
	/* decrease available_resources by count resources */
	/* return 0 if sufficient resources available, */
	/* otherwise return -1 */
	int decrease_count(int count)
	{
	if (available_resources < count)
	return -1;
	else

	{
	available_resources = available_resources - count;
	return 0;
	}
	}
	When a process wants to return a number of resources, it calls the increase_count () function:
	
	/* increase available_resources by count */
	int increase_count(int count)
	{
	available_resources += count;
	return 0;
	}
	The preceding program segment produces a race condition. Do the following:
	a. Identify the location (or locations) in the code where the race condition occurs.
	b. Using a semaphore, fix the race condition.

U18CO031	Write a C program to implement the following game. The parent program P first creates two pipes, and then spawns two child processes C and D. One of the two pipes is meant for communications between P and C, and the other for communications between P and D. Now, a loop runs as follows. In each iteration (also called round), P first randomly chooses one of the two flags: MIN and MAX (the choice randomly varies from one iteration to another). Each of the two child processes C and D generates a random positive integer and sends that to P via its pipe. P reads the two integers; let these be c and d. If P has chosen MIN, then the child who sent the smaller of c and d gets one point. If P has chosen MAX, then the sender of the larger of c and d gets one point. If c = d, then this round is ignored. The child process who first obtains ten points wins the game. When the game ends, P sends a user-defined signal to both C and D, and the child processes exit after handling the signal (in order to know who was the winner). After C and D exit, the parent process P exits. During each iteration of the game, P should print appropriate messages (like P's choice of the flag, the integers received from C and D, which child gets the point, the current scores of C and D) in order to let the user know how the game is going on.
U18CO032	To implement and simulate the MFT algorithm(MEMORY MANAGEMENT WITH FIXED PARTITIONING TECHNIQUE) ALGORITHM: Step1: Start the process. Step2: Declare Variables. Step3: Enter total memory size ms. Step4: Allocate memory for os. Ms=ms-os Step5: Read the no partition to be divided n Partition size=ms/n. Step6: Read the process no and process size.

	Step 7: If process size is less than partition size allocate else block the process. While allocating update memory wastage-external fragmentation.
	if(pn[i]==pn[j])f=1;
	if(f==0){ if(ps[i]<=size)
	{ extft=extft+sizeps[i];
	avail[i]=1; count++;
	 } }
	Step 8: Print the results

U18CO033	write a C program for implement the following algorithm: Step 1: Start the program. Step 2: Get the number of files. Step 3: Get the memory requirement of each file. Step 4: Allocate the required locations to each in sequential order Randomly select a location from availablelocation s1= random(100);
U18CO034	You are given a list of countries, each on a new line. Your task is to read them into an array and then transform them in the following way: The first capital letter (if present) in each element of the array should be replaced with a dot ('.'). Then, display the entire array with a space between each country's names. Write a shell script for that and Print your output.
U18CO035	There is a seminar to be held tomorrow in College. BOB is too anxious to know about his turn, i.e. when he would be presenting on his topic. The turns are allocated on the basis of priority of each individual student. BOB being weak in maths asked Alice for help. Now Alice knows that, The priority of each individual student is defined by the function, F(T) = (M*T) - C , T > 0 = C , T = 0 where, 'M' is the maximum of all initially allocated priorities. 'C' is the current priority of the student. 'T' represents the current time. Just print the correct sequence in which students would be giving their seminars.(take necessary input by yourself).

U18CO036	A Multi-threaded server application stops working and the last log message from the application is:
	"Some Server Related Message..."
	Code looks like:
	CalledFunc ()
	{
	Code ...
	Acquiring Thread lock
	Line printing "Some Server Related Message..."
	Func();
	Releasing Thread Lock
	}
	1. What should the programmer do to debug this?
	2. What has happened wrong in the Func()?

	3. If an exception is thrown in the Func() what should be done to fix the problem ?
U18CO038	Write a shell script program which will perform following: YOUR_SCRIPT -c -l -r -d -e Options work as: -c for clear the screen -l show list of files in current working directory -r remove a file -d copy file -e Rename file