(6 p) The price to parallelize an application whose code is 100% serial is of 30 dollars per percentage unit. That is, if you invest 60 dollars, you can get the serial part of the application to 98%. At the same time, if you want to run such an application on multiple cores, every extra core (aside for the first one) costs 50 dollars. What is better, to invest 210 dollars on parallelizing the code and 100 dollars in paying for cores or rather to invest 120 dollars on parallelizing the code and 200 dollars in paying for cores? Explain why in detail.

Answer hint: do the math using Amdhal's law.

(6 p) Is the following code going to print the right value when executed passing parameter 10 or not? Explain why. If you think there is something to correct, point out all the corrections that must be applied to the code in order to work. Notice that the listing contains only the relevant portions of code, so your answer does not need to take minor inconsistencies (e.g., the missing include statements) into account.

```c
int sum;
void *runner(void *param);

int main(void)
{

   pthread_t tid;
   pthread_attr_t attr;

   if (argc != 2) {
      return -1;
   }

   if (atoi(argv[1]) < 0) {
      return -1;
   }

   pthread_attr_init(&attr);
   pthread_create(&tid,&attr,runner,argv[1]);

   printf("sum is %d\n",sum);

}

void *runner(void *param)
{

   int i, upper = atoi(param), sum = 0;

   for (i=1; i<=upper; i++)
    sum += i;

   pthread_exit(0);
}
```

Answer hint: Nope, no thread join and sum is locally created in runner.

(6 p) Can users use kernel threads? Motivate your answer.

Answer hint: Yes. Everything that is run by the OS is a kernel thread.