1. (10%) Combinations refer to the combination of *n* things taken *p* as a time without repetition. A recursive definition of C(n, p) is C(n, 0) = C(n, n)=1, otherwise C(n, p)=C(n-1, p)+ C(n-1, p-1). Write a function long C(long n, long n) that computes and return C(n, p). Write a program to prompt the user to enter n and p and print C(n, p).
2. (10%, Recursive Prime) Write a recursive function isPrime that determines whether the given input is a prime number. Use this function in a program which ask user to enter a number and check whether the input number is a prime.
3. (20%) The use of computers in education is referred to as computer-assisted instruction (CAI). Write a program that will help an elementary school student learn multiplication. Use the rand function to produce two positive one-digit integers. The program should then prompt the user with a question, such as

How much is 6 times 7?

The student then inputs the answer. Next, the program checks the student’s answer. If it’s correct, display the message “Very good!” and ask another multiplication question. If the answer is wrong, display the message “No, Please try again.” And let the student try the same question repeatedly until the student finally gets it right. A separate function should be used to generate each new question. This function should be called once when the application begins execution and each time the user answers the question correctly.

1. (15%) One problem in CAI environments is student fatigue. This can be reduced by varying the computer’s responses to hold the student’s attention. Modify the program in the previous problem so that various comments are displayed for each answer as follows:

Possible responses to a correct answer:

Very good!

Excellent!

Nice work!

Keep up the good work!

Possible responses to an incorrect answer:

No, Please try again.

Wrong. Try once more.

Don’t give up.

No. Keep trying.

Use random-number generation to choose a number from 1 to 4 that will be used to select one of the four appropriate responses to each correct or incorrect answer. Use a switch statement to issue the response.

1. (15%) More sophisticated computer-assisted instruction systems monitor the student’s performance over a period of time. The decision to begin a new topic is often based on the student’s success with previous topics. Modify the program in the previous problem to count the number of correct and incorrect responses typed by the student. After the student finishes answering a problem and types more than 10 answers, your program should calculate the percentage that are correct. If the percentage is lower than 75%, display “Please ask your teacher for extra help.”, then reset the program so another student can try. If the percentage is 75% or higher, display “Congratulations, you are ready to go to the next level!”, then reset the program so another student can try it.
2. (15%) Modify the program in Problem 4 to allow the user to enter a difficulty level. At a difficulty level 1, the program should use only single-digit numbers in the problems; at a difficulty level of 2, numbers as large as two digits, and difficulty level 3, numbers as large as three digits.
3. (15%) Modify the program in Problem 4 to allow the user to pick a type of arithmetic problem to study. An option of 1 means addition problems only, 2 means subtraction problems only, 3 means multiplication problems only, and 4 means a random mixture of all these types.