763114P ATK I – BASICS OF PROGRAMMING

Exam 15.12.2008

Author of the exam: Jussi Mattas

The maximum number of points is 32. Of these, two (2p) are considered bonus points.

1. (1p each) Explain *briefly*, what the following, programming and C language related concepts mean:
   - compiler
   - assignment statement
   - variable
   - structure
   - programming language
   - array (C language)

2. a) (3p) Describe the structure of a typical C language source code file. How does the execution of the program advance in the source code file? Take into account possible functions (other than `main`).

   b) (3p) How do the main program and other functions communicate with each other?

3. (6p) What is meant by a *pointer* in C language? Give an example of a programming task which requires the use of pointers. Explain why exactly pointers are needed. Comment the following piece of code and determine what the `printf` command would print.

```
int it[3]={2,3,5};
int *ip;
int i=it[2];
ip=it;
*(++ip)=i-1;
printf("%i %i %i\n",i,*ip,*it);
```

4. (6p) How would you represent three-dimensional vectors in C language? Explain at least two ways. How would realize as functions the calculation of the sum and dot product of two vectors?

TURN!

5. a) (4p) Comment carefully the following piece of code. In a few sentences, explain what the function `encrypt` does.

```
void encrypt(char ct[],int n)
{
  int l=strlen(ct);
  int i,j,m=l/n;
  char arr[10][10];
  for(i=0;i<m;i++) {
    for(j=0;j<n;j++) {
      arr[i][j]=ct[j+n*i];
    }
  }
  for(j=0;j<n;j++) {
    for(i=0;i<m;i++) {
      ct[i+m*j]=arr[i][j];
    }
  }
  printf("%s\n",ct);
}
```

What would the following command print?

```
encrypt("whatscookingdoc?",4);
```

b) (4p) Comment all file operations in the following piece of code. In a few sentences, explain what the code does. Would this piece of code be enough for a working main program?

```
FILE *f;
char songs[100][100];
int i;
f=fopen("tracklist.txt","r+");
if(f==NULL) {
  printf("File not found. Exiting...\n");
  return 1;
}
i=0;
fscanf(f,"%s",songs[i]);
while(!feof(f)) {
  printf("%s\n",songs[i]);
  fscanf(f,"%s",songs[++i]);
}
fprintf(f,"end_of_file\n");
fclose(f);
return 0;
```