



DEPARTMENT OF PHYSICS AND ASTRONOMY

Autumn Semester 2010-11

PROGRAMMING IN C

2 Hours

Answer ALL questions

There are a total of 50 marks to be awarded for this paper. The breakdown on the right-hand side of the paper is meant as a guide to the marks that can be obtained from each part of the question.

Please write all answers to questions in your answer booklet and not on the examination paper.

*In the following questions the term “**code fragment**” refers to an incomplete part or section of a computer program.*

TURN OVER

CONTINUED

1. a) What is a library? How are they used? [2]
- b) What is a *conversion character*? What functions are they associated with? [2]
- c) What is a logical operator? Give an example of one. [2]
- d) The statement

```
y = ++x;
```

is an abbreviated way of writing which 2 statements? [2]

e) How many *bits* of memory are used to store a real number in *double* format? [1]

f) What does the *address of* operator do? [2]

g) Write the following *if-else* loop using `?:` the conditional operator instead: [2]

```
if (y>0)
{
    x++;
}
else
{
    x--;
}
```

2. Study the following code fragment and predict the values of **w**, **x**, **y**, **z**, **i**, **j** and **k** at the *printf* statements: [7]

```
double w=0.6397,x=121.667,y=3.02e-3,z=7.46e05;
int i=-4,j=802,k=89;
w = x*y+w/(z-x*x+x);
x /= x-w/z*y-w;
y = (j/k)*x;
z = (z*w-x/i+w)/y-w;
i *= (k%j)-i;
j = i+k*j/i-i;
k += y*(abs(x)%abs(z));
printf ("w:%f x:%f y:%f z:%f\n",w,x,y,z);
printf ("i:%d j:%d k:%d\n",i,j,k);
```

TURN OVER

CONTINUED

3. The following is a short program to calculate the roots of a quadratic equation. Six compilation errors have been introduced into the program, list them: [6]

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
int main(void)
{
// calculate the roots of a quadratic equation

float a,b,c; // the quadratic coefficients
float x1,x2,y1; // roots

printf("Input the coefficients (a,b,c)");
scanf("%f,%f,%f",&a,&b,&c);
// test for complex roots

if (b*b -4.0*a*c >=0.0)
{
// real roots
x1=(-b + sqrt(b*b -4.0*a*c))/(2.0*a);
x2=(-b - sqrt(b*b -4.0*a*c))/(2.0*a);

printf("roots: %f and %f\n",x1,x2);
} else
{
// complex roots
x1=-b/(2.0*a);
y= sqrt(-1.*(b*b -4.0*a*c))/(2.0*a);

printf("roots: %f+i%f and %f-i%f",x1,y1,x1);

}
system(pause);
return 0;
}
```

CONTINUED

4. a) What is dynamic memory allocation? How does it differ from static memory allocation? [3]

b) In the example source code below there are 2 missing statements. Give the syntax required for these 2 lines of code in order to allow the code to dynamically allocate an array of size i (using the *malloc* method) and subsequently free the allocated memory space. [5]

```
#include <stdio.h>
#include <stdlib.h>

int main(void)
{
    int i;
    int* i_array; // this will be the array
    int j;

    // find out how many integers are required
    printf("How many integers? ");
    scanf("%d",&i);

    // Now allocate memory space for the array
    <missing source statement 1>

    // use the new array
    for (j=0;j<i;j++)
    {
        i_array[j]=j;
        printf("%d\n",i_array[j]);
    }
    // free the memory for reuse
    <missing source statement 2>

    system("pause");
    return 0;
}
```

CONTINUED

5. Study the following program carefully and write down exactly what the programme outputs via the *printf* statement.

[8]

```
#include<stdlib.h>
#include<stdio.h>
int main(void)
{
    int i=1, j=10;
    do
    {
        switch (j%6)
        {
            case (1):
            case (2):
                i+=2;
            case (3):
                i+=j-1;
                --j;
                break;
            case (4):
            case (5):
                i++;
            default:
                --j;
                break;
        }
        printf("i: %d j: %d\n",i,j);
    } while (j>4);
    system("pause");
    return 0;
}
```

6. Predict the output produced by the following code fragment. Make leading and trailing spaces clear in your answer by using an underscore symbol ().

[8]

```
int StarCatNum = 7089;
double StarRightAsc = 37.5211;
char StarName[9] = "Achernar";
//
printf("%-4d\n",StarCatNum);
printf("%+2d\n",StarCatNum);
printf("%f\n",StarRightAsc);
printf("%7.2f\n",StarRightAsc);
printf("%6.3e\n",StarRightAsc);
printf("%-.5s\n",StarName);
printf("%-10.8s\n",StarName);
printf("%7.3s\n",StarName);
```

END OF QUESTION PAPER