

MCS-011 Solved Assignment

1. Mention the type of applications which can be developed using C language. Also, list the latest C compilers that are available in the market by giving the complete details like the name of the compiler, version, vendor's name, DOS based /WINDOWS based / Others, year of release, etc..
(10 Marks)

Ans:

- UNIX kernel is completely developed in C Language.
- Operating Systems
- Network Drivers
- Print Spoolers
- Language Compilers
- Assemblers
- Text Editors
- Modern Programs
- Data Bases
- Language Interpreters
- Simulators
- Utilities
- Embedded System

List of Compilers

- **DeSmet C**: DeSmet C will be familiar to those who programmed in C in the 1980s. It is an MSDOS C compiler. It has been released under the GNU GPL, and comes with manuals, an editor, and a third party optimizer.
- **Ch Embeddable C/C++ Interpreter (Standard Edition)**: Ch is a C/C++ interpreter that supports the ISO 1990 C Standard (C90), major features in C99 (complex numbers, variable length arrays or VLAs, type generic functions, the "long long" data type, etc), classes in C++, and extensions to the C language like nested functions, string type, etc. It can be embedded in other applications and hardware and used as a scripting language. Your C/C++ code is interpreted directly with no compilation to intermediate code. Since it supports [Linux](#), Windows, MacOS X, Solaris and HP-UX, it means that your code should be portable to any of those platforms once you write it for this compiler. The standard edition is free for personal, academic and commercial use. You need to register to download the package.
- **SDCC C Cross-compiler**: This is a C cross-compiler that targets the Intel 8051, DS390, Z80, HC08 and PIC microprocessors. It can also be retargeted for other 8 bit MCUs or PICs. It comes with a retargetable assembler and linker, a source level debugger and a simulator, and it is capable of a variety of optimisations. The libraries are Standard C99 compatible. Source code for the compiler is available under GPL. Host platforms supported include Linux, Windows, Mac OS X, Alpha, Sparc, etc.
- **LCC - A Retargetable Compiler for ANSI C**: LCC is a C compiler (source code only) that generates code for the Alpha, Sparc, MIPS R3000 and Intel x86. There is also a book [A Retargetable C Compiler](#) written by the authors of the compiler that explains the code of the C compiler. This compiler is the basis of at least 2 other Win32 C compilers
- **Cyclone C**: Cyclone C is not strictly an ANSI C compiler but a compiler of a "safe dialect" of C. It enforces type safety, has a variety of checks to protect against buffer overflows, array-bound violations, etc. It currently works on [Linux](#) and Windows (the latter via Cygwin), and requires you to have the GNU compiler tools on your system

2. Draw a flow chart and write a corresponding interactive program which prompts the user with the following options on the opening menu: 1) Subtract two integers 2) Compare two integers to find the

smallest 3) Test an integer for odd or even 4) Quit Enter your choice: If an "1" is entered, prompt for the input of two integers and display their difference. If "2" is entered, prompt for two integers and display the smaller of the two. If "3" is entered, prompt the user for one integer and print out if it is odd or even. If "4" is entered, exit the program. If the user enters any letters or numbers other than the choice, redisplay the prompt. All output should go to the terminal and all input should come from the keyboard. (10 Marks)

Ans:

```
#include<stdio.h>
#include<math.h>
main()
{
int a,b,c,e;
clrscr();
do
{
printf("-----Menu-----\n");
printf("1- for Subtract two integer\n");
printf("2- for Compare two Integer\n");
printf("3- for Add \ Even checks\n");
printf("4- for Exit\n");
printf("-----Menu-----\n");
printf("enter your choice :");
scanf("%d",&e);
switch(e)
{
case 1:
printf("enter he two umbers :");
scanf("%d%d",&a,&b);
c=a-b;
printf("Subtraction=%d",c);
break;
case 2:
printf("enter he two umbers :");
scanf("%d%d",&a,&b);
if(a==b)
printf("Both are equal");
else
printf("Both are not equal");
break;
case 3:
printf("enter a number");
scanf("%d",&a);
if(a%2==0)
printf("even");
else
printf("Odd");
break;
case 4:
break;

default:
```

```
printf("\nwrong choice\n");
}
printf("\nDo you want to continue(1-YES, 0-NO):");
scanf("%d",&c);
}while(c!=0);
getch();
}
```

3. Write an interactive program using strings which: a) gets a filename from the standard input (keyboard) or a file b) gets a mode (read or write) from the same source and, c) copies the contents of the input file to: (i) the standard output if the input is from a file OR (ii) to the file specified in a) if the mode from b) is write If the file won't open, direct the input/output to the corresponding standard file (stdin/stdout). (10 Marks)

Ans:

```
#include<stdio.h >

main()
{
FILE *fp,*pp;
char ch,*s;

clrscr();
printf("enter the File name");
gets(s);
fp=fopen(s,"r"); //reopen file for reading.
pp=fopen("copy.txt","w");

if(fp==NULL)
{
printf("file can not be open");
exit();
}

while(ch!=eof())
{
ch=fgetc(fp);
putchar(ch);
fputc(ch,pp);
}
```

```
}
getch();
}
```

4. Draw a flowchart and write an interactive C program that prints a *power* table for a specified range of integers. The user specifies the starting and ending integer on the command line along with the max power to compute for each integer. An example is included below: Example: Starting Integer: 2 Ending Integer: 4 Maximum Power to be computed: 5 Output Num Powers (1 - 5) 2 2 4 8 16 32 3 9 27 81 243 729 4 16 64 256 1024 4096 Your program should use the *pow()* function along with casting of this function's arguments and output. (20 Marks)

Ans:

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

```
#include<math.h>
```

```
main()
```

```
{
```

```
int p,s,e,i,j,ot;
```

```
clrscr();
```

```
printf("enter starting number");
```

```
scanf("%d",&s);
```

```
printf("enter the ending number");
```

```
scanf("%d",&e);
```

```
p=s-1; //starting position
```

```
for(j=s;j<=e;j++)
```

```
{
```

```
for(i=p;i<=e+1;i++)
```

```
{
```

```
ot=pow(s,i);
```

```
printf("%d\t",ot);
```

```
}
```

```
s++;
```

```
printf("\n");
```

```
}
```

```
getch();
```

```
}
```

5. Write an interactive C program to simulate the evaluation scheme for MCA (First semester) for 10 students. Each course should have both the components (Assignment as well as Term End Examination). (10 Marks)

Ans:

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

```
struct student
```

```
{
```

```
    int roll;
```

```
    char name[50];
```

```
    int sub[7],assign[7];
```

```
};
```

```
void main()
```

```
{
```

```
    int k,i;
```

```
    struct student s[10];
```

```
    clrscr();
```

```
    for(i=0;i<10;i++)
```

```
    {
```

```
        printf("Enter name: ");
```

```
        scanf("%s",s[i].name);
```

```
        printf("Enter roll number: ");
```

```
        scanf("%d",&s[i].roll);
```

```
    for(k=0;k<=6;k++)
```

```
    {
```

```
        printf("Enter Term-End Marks of Sub %d=",k+1);
```

```
scanf("%d",&s[i].sub[k]);

printf("Enter Assignment of Sub %d=",k+1);

scanf("%d",&s[i].assign[k]);

}

}

printf("\nDisplaying Information\n");

for(i=0;i<10;i++)

{

printf("\n_____ \n");

printf("Roll:%d \t Name:%s \n",s[i].roll,s[i].name);

printf("\n_____ \n");

for(k=0;k<=6;k++)

{

printf("TTE MARKs:%d \t Assignment:%d\n", s[i].sub[k],s[i].assign[k]);

}

}

getch();

}
```

6. Write a program to *crypt* its input according to a specified transformation scheme. The transformation scheme will consist of two strings: a string of characters and then a string of replacement characters. The idea is that your program replaces every instance of the i th character in the initial string with the $(i+1)$ th character (of English alphabets) in the replacement string. It follows a cyclic pattern. If alphabet *z* is met it starts with alphabet *a*. When no substitution is defined for a character, the program just passes it through to the output unchanged. Blank spaces and the other symbols remains the same. The program should inform the user of any errors in the transformation scheme. Your program should display the strings before and after the substitutions in the corresponding 2 files named *bcrypt* and *acrypt*. Example: Original String: I know C programming. String after the transformation: J lopx D qsphsbnnjoh.

Ans:

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

```
main()
```

```
{
```

```
FILE *fp,*pp,*op;

char ch,*s;

clrscr();

op=fopen("acrypt.txt","w+r");

fp=fopen("bcrypt.txt","w");

if(fp==NULL)

{

printf("file can not be open");

exit();

}

printf("enter the Text");

gets(s);

printf("\nBefore the transformation=%s\n",s);

while(*s!=NULL)

{

fputc(*s,op);

s++;

}

rewind(op);

printf("\nAfter Transformation=");

do

{

ch=fgetc(op);

if(ch>='a' && ch<='z')

ch=ch+1;

putchar(ch);

fputc(ch,fp);
```



```
}while(ch!=eof());
```

```
getch();
```

```
}
```



**HOW TO PREPARE IGNOU MCA SUBJECTS IN ONE DAY,
IGNOU IMP QUESTIONS, IGNOU TOPICS VIDEO LECTURES
ALL ARE AVAILABLE ON OUR YOUTUBE CHANNEL.....JUST
"CLICK HERE"**