Indian Institute of Technology, Patna

End Semester Examination

CS101: Introduction to Computing

Full Marks: 100 Time: 3hours Date: 27.04.2016

Name:	**Roll No.:**	**Your Signature:**	**Invigilators Signature:**

Q1. State the output of the following codes. Assume no syntax error exists (12 x 2.5 =30)

a.

```
#include<stdio.h>
fun1(int a[ ], int nrow)
{
inti,j,t;
for(i=1;i<nrow;i++)
    for(j=i;j>0;j--)
         if(a[j]<a[j-1])
                  {
                  t=a[j];
                  a[j]=a[j-1];
                  a[j-1]=t;
                  }
         else break;
}

main()
{
inti;
int a[]={3,1,9,7,2,8,5};
fun1(a, 7);
for(i=0;i<7;i++) printf("%d ", a[i]);
}
```

Output: ________________________________

____

b.

```
#include <stdio.h>
void main()
{
int num, x, y, count = 0;
          num=85;
  x = num << 1;
  y = x ^ num;
  y = y + 1;
  while ((y / 2) != 0)
  {
    if (y % 2 != 0)
    {
      count++;
      break;
    }
else y = y / 2;
  }
  if (count) printf("false");
  else printf("true");
}
```

Output: ________________________________

____

c.

```
#include <stdio.h>
#include <string.h>
void check(char word[ ], int
index)
{
int len = strlen(word) - (index +
1);
  if (word[index] == word[len])
  {
    if (index + 1 == len || index
== len)
     {
printf("TRUE");
       return;
     }
 check(word, index + 1);
  }
  else  printf("FALSE");
```

```
int main()
{
  char word[15];
strcpy(word,"mantain"
);
  check(word, 0);
  return 0;
}
```

d.

```
#include<stdio.h>
#include<stdlib.h>
void fun(int a, int* b, int m)
{
inti, *c,k, q=0;;
c=(int*)malloc((m+1)*sizeof(int));
for(i=0;i<=m;i++)  c[i]=0;
for(i=0;i<a;i++) c[b[i]]++;
for(i=1;i<=m;i++) c[i]+=c[i-1];
for(i=0;i<=m;i++)
  {
  if(i==0) k=c[i]; else k=c[i]-c[i-1];
   while(k!=0) { b[q]=i; q++; k--;}
  }
}
```

```
main()
{
intarr[5]={2,3,0,3,2};
inti;
fun(5,arr,3);
for(i=0;i<5;i++)
printf("%d\t",arr[i]);
}
```

Output:

____

```
}
Output:
```

____

–

____

____

e.

```
#include <stdio.h>
#define PI 3.1415
#define area(r) (PI*r*r)
int main(){
int radius;
  float area;
  radius=5;
area=area(radius+1);
printf("%.2f",area);
  return 0;
}
```

Output: ______________________

____

f.

```
typedef struct structc_tag
{
  char c;
  double d;
int  s;
} structc_t;

typedef struct structd_tag
{
  double d;
int  s;
  char c;
} structd_t;
```

```
int main()
{
printf("sizeof(structc_t)=%d\n
", sizeof(structc_t));
printf("sizeof(structd_t)=%d\n
", sizeof(structd_t));
  return 0;
}
```

Output:

____

____

g.

```
#include<stdio.h>
int fun(intnum)
{
  if (num == 0)  return 0;
  else if (num == 1)  return 1;
  else
  {
    return(fun(num  -  1)  +
fun(num - 2));
  }
}
```

```
int main()
{
intnum;
int result;
  num=6;
  result = fun(num);
printf("%d ", result);
  return 0;
}
```

Output:

____

____

h.

```
#include<stdio.h>
int main(){
inti=-3, j=2, k=0, m;
m= ++i&& ++j && ++k;
printf("%d %d %d %d", i, j, k, m);
return 0;
}
```

Output: ______________________

____

i.

```
#include<stdio.h>
int main()
{
        int fun(int);
        inti=3;
        fun(i=fun(fun(i)));
        printf("%d", i);
        return 0;
}

int fun(inti)
{
```

j.

```
#include<stdio.h>
int main()
{
int i= 1;
for(; ;)
        {
        printf("%d", i++);
        if(i>10) break;
        }
return 0;
}
```

```
        i++;
        return i;
}
```

Output: ____________________

____

____

k.

```
#include<stdio.h>
void fun(int);
int main()
{
int a;
a=3;
fun (a);
printf("\n");
return 0;
}
```

Output:

____

____

```
void fun (int n)
{
if (n>0)
{
fun (--n);
printf("%d",n);
fun (--n);
}
}
```

l.

```
#include<stdio.h>
int main()
{
int a[5]={5,1,15,20,25};
inti,j,m;
i=++a[1];
j=a[1]++;
m=a[i++];
printf("%d%d%d\n",i,j,m);
return 0;
}
```

Output: ____________________

____

Q2. State the error in the following codes. (12 x 2.5 =30)

a.

```
int fun(int a[ ][ ], int nrow, int ncol)
{
inti, j;
for(i=0;i<nrow;i++)
        for(j=0;j<ncol;j++)
                a[i][j]*=a[j][i];
}

main()
{
int a[3][3]={{0,1,1},{1,0,1},{1,1,0}};
inti,j;
fun(a,3,3);
for(i=0;i<3;i++)
        {
        for(j=0;j<3;j++)
                printf("%d\t", a[i][j]);
        printf("\n");
        }
}
```

Error:

b.

```
#include<stdio.h>
main()
{
int a[5],i;
char c[5]={'a','b','c','d','e'};
for(i=0;i<5;i++)
 a[i]++=c[i];
}
```

Error:

c.

```
#include<stdio.h>
fun(a[ ][3], int nrow, intncol)
{
inti,j, sumR[3];
for(i=0;i<nrow;i++)
        sumR[i]=0;
```

d.

```
#include<stdio.h>
typedef struct node{
int item;
struct node next;
}node;
```

```
for(i=0;i<nrow;i++)
        for(j=0;j<ncol;j++)
        {
        sumR[i]+=*(*a[i]+j);
        }
for(i=0;i<nrow;i++)
        printf("\n%d", sumR[i]);
}

main()
{
inti,j;
int a[][3]={{1,1,1}, {2,2,2},{3,3,3}};
fun(a,3,3);
}
```

Error:

```
main()
{
struct node *firstNode, *lastNode;
int count=5;
firstNode=malloc(sizeof(node*));
firstNode->item=2;
firstNode->next=NULL;
lastNode=firstNode;
while(count>0)
        {
        lastNode->next=malloc(sizeof(node*));
        lastNode=lastNode->next;
        lastNode->item=2;
        lastNode->next=NULL;
        count--;
        }
}
```

Error:

e.

```
#include<stdio.h>
void love(char (*arr)[5][5])
{inti, j;
for(i=0;i<5;i++)
   for(j=0;j<5;j++)
         printf("%c",(*arr)[i][j]);
}
int main()
{char arr[5][5];
inti,j;
 for(i=0;i<5;i++)
   for(j=0;j<5;j++)
         arr[i][j]='*';
 love(arr);
}
```

Error:

f.

```
#include <stdio.h>
#include<string.h>
int main()
{
char c[1000],d[1000];
strcpy(c,"an");
strcpy(d,"man");
d++;
if(c==d)
printf("Sub String");
return 0;
}
```

Error:

g.

```
#include<stdio.h>
intmain()
{
struct xx
  {
int x=3;
    char name[]=“hello”;
  }
struct xx *s=malloc(sizeof(struct xx));
printf(“%d”,s->x);
printf(“%s”,s->name);
  return 0;
}
```

Error:

h.

```
#include<stdio.h>
intmain()
{
  static char names[5][20] =
{“pascal”,“ada”,“cobol”,“fortran”,“perl”};
int i;
   char *t;
   t=names[3];
   names[3]=names[4];
   names[4]=t;
   for (i=0;i<=4;i++)
printf(“%s”,names[i]);
   return 0;
}
```

Error:

i.

```
#include<stdio.h>
int main()
{
struct xx
 {
          int x;
          struct yy
          {
          char s;
          struct xx *p;
          }
          struct yy *q;
}
   return 0;
}
```

Error:

j.

```
#include<stdio.h>
Int main( )
{
int a[ ] = {10,20,30,40,50},j,*p;
for(j=0; j<5; j++)
{
printf(“%d” ,*a);
a++;
}
p = a;
for(j=0; j<5; j++)
{
printf(“%d ” ,*p);
p++;
    }
    return 0;
 }
```

Error:

k.

```
#include<stdio.h>
int main()
{
inti=1,j=2;
switch(i)
{
case 1:  printf(“GOOD”);
break;
case j:  printf(“BAD”);
break;
}
   return 0;
}
```

Error:

l.

```
#include<stdio.h>
int main()
{
inti=5;
printf(“%d”,i+++++i);
     return 0;
}
```

Error:

Q3. Complete the missing code parts according to the given objective. Assume there are no syntax errors in the codes (6 x 5 =30)

Write the program to sort the elements in an array in descending order:

```
#include<stdio.h>
#include<conio.h>
void main()
{
intarr[5]={5, 6, 2, 8, 1};
inttemp, i, j, n;
for(i=0;i<5;i++)
{
for(j=i+1; j<5; j++)
{
if(_______________________________)
{
```

Write a program for finding the largest number in an array

```
#include<stdio.h>
#include<conio.h>
void main()
{
int *arr, i, j, n, LARGE;
printf(“Enter the number of elements”);
scanf(“%d”, &n);
arr=(int*) malloc(sizeof(int)*n);
for(i=0; i<n; i++)
{
printf(“Enter a number”);
scanf(“%d”, &arr[i]);
}
```

```
 temp=arr[i];
 arr[i]=arr[j];
 arr[j]=____________________;
 }
    }
 }
printf(“Elements in descending order are”);
for(i=0; i<5; i++);
printf(“%d”, a[i]);
}
```

```
LARGE=arr[0];
for(i=1; i<n;i++)
{
if(____________________)
LARGE=arr[i];
}
printf(“The largest number is : %d”, LARGE);
}
```

```
/* Given a stack of n disks arranged from largest on
the bottom to smallest on top placed on a rod,
together with two empty rods, the towers of Hanoi
puzzle asks for the minimum number of moves
required to move the stack from one rod to another,
where moves are allowed only if they place smaller
disks on top of larger disks.
*/

#include<stdio.h>

void tower(int ,char ,char ,char );

void main()
{
int n;
clrscr();
printf("\n\n Enter number of disk");
scanf("%d",&n);
  tower(n,'S','A','D');
getch();
}

void tower(intN,charS,charA,char D)
{
  if (N== 1)
printf("\nMove from tower %c to tower %c",S,D) ;
  else
  {
        tower(____________________);
        printf("\nMove from tower %c to tower %c",
S,D) ;
        tower(____________________);
  }
}
```

```
/*C Program to Find the Frequency of Substring in the given
String */
#include <stdio.h>
#include <string.h>
void main()
{
  int count = 0, i, j = 0, k;
  char str[100], str1[20];
  printf("Enter the string\n");
  gets(str);
  printf("Enter substring to be matched\n");
  gets(str1);
  k = strlen(str1);
  for (i = 0;____________________;)
  {
    if (str[i] == ' ')  i++;
    else
    {
      if (str[i] == str1[j])
      {
        j++;
        i++;
      }
      else if (j == k)
      {
        j = 0;
        ____________________

        ____________________
      }
      else
      {
       ____________________



       ____________________
      }
    }
  }
  printf("No of matches of substring in main string is %d\n",
count);
}
```

/* C Program to Reverse the Contents of a File and Print it in a different file. The filenames are passed as a command line argument, where the first file name is the file to read from and the next one is the file to write.

```
/*Program to find whether a Number is Prime or Not using
Recursion */

#include <stdio.h>
int primeno(int num, int i)
```

```
 */
#include <stdio.h>
#include <errno.h>
/* This function counts the total number of characters
in the file that *f points to*/

long count_characters(FILE *f)
{

  ______________________
  long last_pos = ftell(f);
  last_pos++;
  return last_pos;
}

void main(int argc, char * argv[])
{
  int i;
  long cnt;
  char ch, ch1;
  FILE *fp1, *fp2;
  if (fp1 = fopen(______________, "r"))
  {

    fp2 = fopen(______________, "w");
    cnt = count_characters(fp1);

    ______________________
    while (cnt)
    {
      ch = fgetc(fp1);
      fputc(ch, fp2);

      ______________________
      cnt--;
    }
  }
  else
  {
    printf("File Open Error\n");
    exit(1);
  }
  fclose(fp1);
  fclose(fp2);
}
```

```
{
  if (______________)    return 1;

  else
  {
   if (________________) return 0;


   else
     return ______________________
  }
}


int main()
{
  int num, check;
  printf("Enter a number: ");
  scanf("%d", &num);
  check = primeno(num, num / 2);
  if (check == 1) printf("PRIME);
  else printf("NOT PRIME");
  return 0;
}
```

24. Write the following functions. Use Supplementary sheets for the same. (2 x 5 =10).

1. Given a linked list where each node contains two fields: (i) **data**: to store integer value and (ii) **next**: to point to next node. Let h be a pointer which points to the first node in the linked list. Write a function which takes&h as parameter and returns the second largest value in the linked list.
2. Write a function HCF(int a, int b) that returns the highest common factor of the numbers a and b.