Department of Computer Science and Engineering
Indian Institute of Technology Patna

Course:CS101 — End Sem Exam

Time: 180 minutes — Full Marks 100

Roll No: ______________ Signature: ____________

Name: ______________________________________

Invigilator's Signature: ____________________

Answer in the question paper itself. Rough work can be done in the supplementary sheets.

1. Write down a correct output for each of the following C programs. Give proper justification for your answers. Assume that all the necessary header files are included and there is no syntax error.
   (12 × 2) = 24

(a)
```
void main(){
       char *str="Hello world";
       printf("%d",printf("%s",str));
   }
```

Answer:

____

Justification:______________________________

____

(b)
```
union
     {
       int ival;
       float fval;
     } u;
void main(){
     printf("%d", sizeof(u));
     }
```

Answer:

____

Justification:______________________________

____

(c)
```
void main(){
       int x,y = 1;
       if (y & (x = 2))
         printf("true %d\n", y);
       else
         printf("false %d\n", x);
     }
```

Answer:

____

Justification:______________________________

____

(d)
```
enum colors {BLACK,BLUE=4,GREEN,WHITE=10}
int main(){
    printf("%d..%d..%d",BLACK,WHITE,GREEN);
    return(1);
  }
```

Answer:

____

Justification:______________________________

____

(e)
```
#define prod(a,b) a*b
void main(){
    int x=3,y=4;
    printf("%d",prod(y+2,x-1));
  }
```

Answer:

____

Justification:______________________________

____

(f)
```
void main(){
    printf("%x",10<<2);
  }
```

Answer:

____

Justification:______________________________

____

(g)
```
int main(){
  int A[]={3,2,1},r=1, i;
  for(i=0;i<=2;++i)
   {
    if(A[i]>A[r])
     r=i;
   }
  printf("%d",r);
  return 0;
 }
```

Answer:

____

Justification:______________________________

____

(84)

(h)
```
int main(){
    int A[]={3,2,1}, sum, biggest, i;
    float average;
    for(i=0, sum =A[0], biggest=sum; i<3;
                            sum=sum + A[i++])
        if(A[i+1]>biggest)
         biggest=A[i+1];
    average = sum/3;
    printf("%d %.2f",biggest, average);
    return 0;
  }
```

Answer:

____

Justification:____________________________

____

(i)
```
int main(){
    int A[]={10,20,30,40,50};
    int *p;
    p = A;
    printf("%d %d",*p+4, *(p+4));
    return 0;
  }
```

Answer:

____

Justification:____________________________

____

(j)
```
#define SQR(x) (x*x)
void main(){
  int p=2, res;
  res=SQR(p+1);
  printf("The result is %d\n", res);
 }
```

Answer:

____

Justification:____________________________

____

(k) (For this problem assume a 32 bit machine)

```
struct account
{
 char name[20];
 char num[17];
 int ledgerNo;
 int balance;
};
void main(){
 struct account myacc;
 printf(Size of myacc structure is %d bytes
                      \n, sizeof(myacc));
}
```

Answer:

____

Justification:____________________________

____

(l)
```
void fun(int a, int* b, int m)
{
 int i, *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--;}
    }
}

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

Answer:

____

Justification:____________________________

____

2. For each of the following program/program segment/functions, point out if there is any error. Give proper justification for your answers. Assume that all the necessary header files are included.
$(12 \times 2) = 24$

(a)
```
#define max 10
void main(){
 int i=0;
 i=max++;
 printf("%d",i++);
}
```

Answer:

____

Justification:____________________________

____

(b)
```
int main(){
    int a=0;
    #if (a==0)
      printf("Equal");
    #else if
      printf("Not equal");
    #endif
    return 0;
}
```

Answer:

____

Justification:______________________________

____

(c)
```
void main(){
    int a=2;
    if(a==2){
      a=~a+2<<1;
      printf("%d",a);
    }
    else{
      break;
    }
  }
```

Answer:

____

Justification:______________________________

____

(d)
```
void main(){
  register int a=2;
  printf("Address of a = %d",&a);
  printf("Value of a   = %d",a);
    }
```

Answer:

____

Justification:______________________________

____

(e)
```
int main(){
      FILE *ptr;
      char i;
      ptr = fopen("sample.c", "r");
      while((i=fgetc(ptr))!=NULL)
          printf("%c", i);
      return 0;
  }
```

Answer:

____

Justification:______________________________

____

(f)
```
int main(){
       int a = 5;
       switch(a)
       {
       case 1: printf("First");

       case 2: printf("Second");

       case 3 + 2: printf("Third");

       case 5: printf("Final");
       break;
       }
       return 0;
   }
```

Answer:

____

Justification:______________________________

____

(g)
```
int main(){
     int i=0;
     switch(i)
      {
          case 0: i+=1;
          case 1: i+=2;
          default: i+=4;
          break;
          printf("%d ",i);
       }
     return 0;
   }
```

Answer:

____

Justification:______________________________

____

(h)
```
union Data
   {
      int i;
      char str[5];
   };
   int main(){
      union Data d;
      d->i=0;
      printf("size of union %d ", sizeof(d));
      return 0;
   }
```

Answer:

____

Justification:______________________________

____

(i) Consider the following program to match two strings. What is the problem in this code?

(85)

```
int main(){
  char str1[]="IIT Patna";
  char str2[]="IIT Patna";
  if(str1 == str2)
     printf("Matched");
  else
     printf("Not Matched");
  return 0;
}
```

Answer:

____

Justification:______________________________

____

(j)
```
typedef struct node
{
 int elem;
 node *next;
}node;

void main(){
 node *ball1, *ball2;
 ball1->elem=5;
 ball2->elem=6;
 ball1->next=ball2;
 ball2->next=ball1;
}
```

Answer:

____

Justification:______________________________

____

(k)
```
void main(){
 int a[5],i;
 char c[5]={'a','b','c','d','e'};
 for(i=0;i<5;i++)
 a[i]++=c[i];
}
```

Answer:

____

Justification:______________________________

____

(l)
```
void fun(int a, int b)
{
 int i;
 for(i=0;i<4;i++)
    a[i]=b[i]++;
}
void main(){
 int a[10], b[4]={1,4,6,7};
 int i;
 fun(a, b);
 for(i=0;i<4;i++)
    printf("%d\t", a[i]);
}
```

Answer:

____

Justification:______________________________

____

3. Complete the C function/program for each of the followings. Fill in the blanks only. (4 × 6) = 24

(a) The following program prints the number of trailing zero(s) in the binary representation of any unsigned integer.

```
void main(){
 int count = 0;
 unsigned int num;
 printf("enter the number:");
 scanf("%d", &num);
 while (num != 0)
  {
   if (num & 1 == 1)

        ______________________;
   else {
         _____________________;

         _____________________;
        }
  }
  printf("\n%d", count);
}
```

(b) The following program prints the Highest Common Factor (H.C.F) of two numbers.

```
int main()
{
    int num1,num2;
    printf("Enter two integers: ");
    scanf("%d %d",&num1,&num2);
    printf("HCF of %d and %d is ",num1 , num2);

    while(__________________)
    {
        if(num1>num2)
            num1=_______________;
        else
            num2-=num1;
    }
    printf("%d",num1);
    return 0;
}
```

(c) Complete the following function int *lastOccur(char *str, char ch)* which takes a string *str* and a character *ch* as input and it returns the position of the last occurrence of the character *ch* in *str* or it returns -1 if the character *ch* is not present in *str*. Do not use any extra variable or any string library function. Each blank indicates that you need to add one statement/condition.

```
int lastOccur(char *str, char ch)
{
  int i = 0;
 /*Go to the end of str*/

while(___________________________)

     _________________________________;
/*From the end of string traverse towards
                           the beginning*/

while(_____________________________)
/*Check if ch occurs in the string*/

  if(______________________________)
     return i;
  return -1;
}
```

(d) The objective is to multiply matrix a[10][10] with b[10][5], store the result in c[][]and print the same in proper format

```
void matrixMultiply(int a[][10], int b[][5])
{
int c[10][5], i,j,k;
//This loop is for initialization of c[][]

for(i=0;______________;i++)

  for(j=0;____________;j++)
              c[i][j]=0;

//This loop is for actual multiplication
for(i=0;i<10;i++)
   for(k=0;k<5;k++)

   ____________________________

              ____________________________

//This loop is for printing

for(i=0;_______________;i++)
  {
    for(j=0;_______________;j++)
    printf("%d   ", c[i][j]);

         __________________________

      }

   }
```

4. Write a C program or function as stated in the followings- (4 × 7) = 28

   (a) Write a complete C program which will print itself (the full code) as output. You can not use the file name as a string in your program.

   `Answer:`

(86)

(b) Write a function named *"hydroxide"* that returns *'1'* if its string argument ends in the substring *"OH"*.

Answer:

(c) Write a function that accepts as an argument an array of characters together with its size n and non-negative integer k. The function should return another array, allocated dynamically within the function, which is obtained by cyclically shifting the input array A by k position to the right. For example, upon the input of $A=$'a','b','c','d','e' of size n=*5* and k=*2*, the function should return 'd','e','a','b','c'.

Answer:

(87)

(d) Write a function with prototype, *node* represent-Polynomial(int a[], int maxPower)* that takes as input an array consisting of the coefficients of a polynomial and represents the non-zero coefficients and corresponding power of x using a linked list of struct node. The index of the array indicates the corresponding power of x. *maxPower* represents the maximum power of x of the polynomial. The return type is a structure node defined as

```
typedef struct node{
            int coeff;
            int exp;
            struct node *next;
            } node;
```

Thus a polynomial $2+3x-4x^3$ will be represented as *[2,3,0,-4]* in the array and linked list entries would be (2,0)→(3,1)→(-4,3).
Only write the required function, not the entire program.

Answer: