C-DAC/ACTS

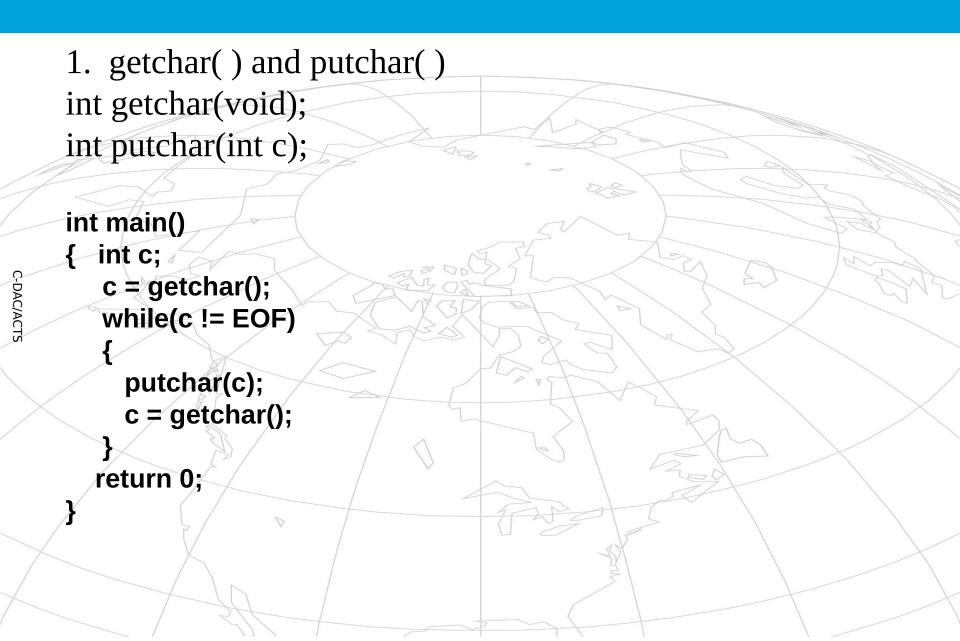


C Programming Concepts

Kaushal Kishor Sharma
Advanced Computing Training School (ACTS)

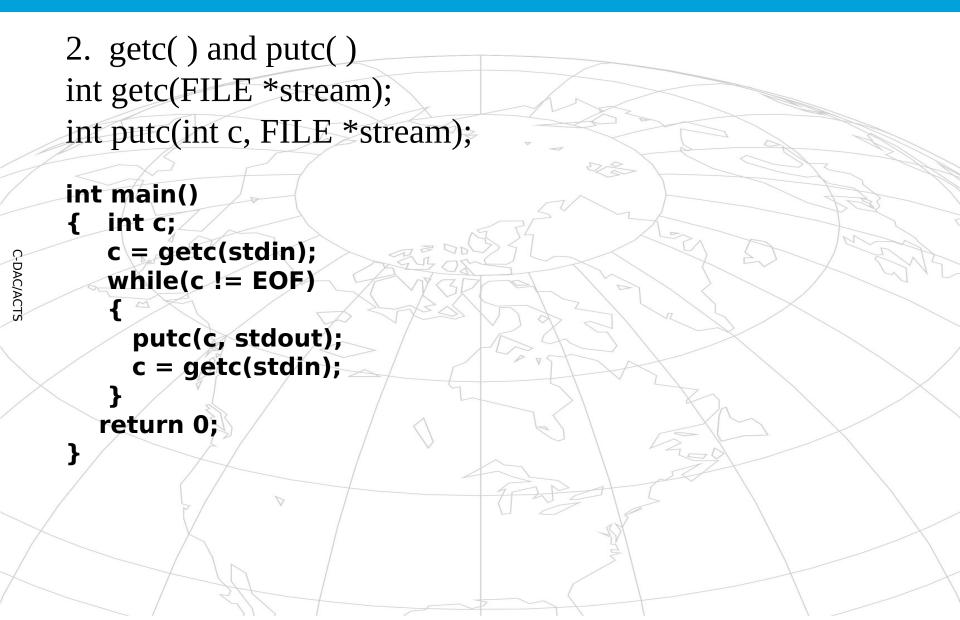
CDAC, Pune
Kaushal.comp@gmail.com















```
3. fgetc() and fputc()
int fgetc(FILE *stream);
int fputc(int c, FILE *stream);
int main()
   int c;
   c = fgetc(stdin);
   while(c != EOF)
     fputc(c, stdout);
     c = fgetc(stdin);
  return 0;
```



```
4. gets() and puts()
    char *gets(char *s);
    int puts(const char *s);
    int main()
C-DAC/ACTS
          char arr[10];
          char *str;
                        // char *str=arr;
          gets(arr);
          gets(str);
          puts(arr);
          puts(str);
```



```
5. fgets() and fputs()
char *fgets(char *s, int size, FILE *stream);
int fputs(const char *s, FILE *stream);
int main()
     char arr[10];
     char *str;
     str=fgets(arr, 10, stdin);
     fputs(str, stdout);
     fputs(arr, stdout);
```

```
6. fscanf() and fprintf()
int fscanf(FILE *stream, const char *format, ...);
int fprintf(FILE *stream, const char *format, ...);
int main()
     int x;
     char ch;
     fscanf(stdin,"%d %c", &x, &ch);
     fprintf(stdout, "%d %c\n", x, ch);
```

C-DAC/ACTS



```
7. sscanf() and sprintf()
int sscanf(const char *str, const char *format, ...);
int sprintf(char *str, const char *format, ...);
int main()
     char *input="hello 100 3.5";
     char outstr[20];
     char str[10]; int i; float f;
     sscanf(input, " %s %d %f", str, &i, &f);
     printf("%s %d %f\n", str, i, f);
     sprintf(outstr, "%f %d %s", f, i, str);
     puts(outstr);
```



```
8. read() and write()
  ssize_t read(int fd, void *buf, size_t count);
  ssize_t write(int fd, const void *buf, size_t count);
  int main()
       char buf[30];
C-DAC/ACTS
       int size;
       size=read(0, buf, 10);
       write(1,buf, size);
```

fread(buf, 1, 10, stdin);

fread(buf1, 4, 5, stdin);

fwrite(buf, 1, 10, stdout);

fwrite(buf1, 4, 5, stdout);

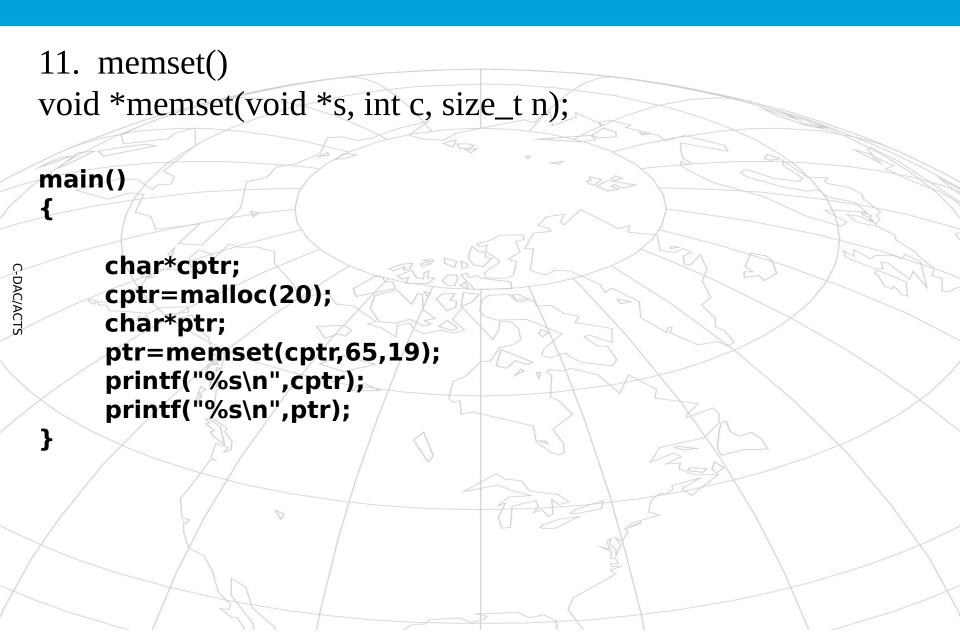


Advanced Computing Training School (ACTS)



```
10
  int main()
       printf("enter the strings:");
       char a[400], ch;
       char *sptr=a;
       int count=0;
       while((ch=getchar())!=EOF)
                           count++;
            *sptr++=ch;
C-DAC/ACTS
       *sptr='\0';
       count=0;
       sptr=a;
       while((*sptr)!='\0')
            putchar(*sptr);
            sptr++; count++;
```







```
12. memcpy()
  void *memcpy(void *dest, const void *src, size_t n);
  main()
       char a[20],b[20];
       char *ptr;
C-DAC/ACTS
       scanf("%s",b);
       ptr=memcpy(a,b,10);
       printf("%s\n",a);
       printf("%s\n",ptr
```



```
13. memcmp()
  int memcmp(const void *s1, const void *s2, size_t n);
  main()
       char ch='a';
       int i=97;
C-DAC/ACTS
       char *ptr1=&ch;
       int *ptr2=&i;
       int flag=memcmp(ptr1,ptr2,1);
       printf("%d\n",flag);
```



```
void g_swap(void*,void*,int );
main()
    int a=10,b=20;
    char c1='a',c2='b';
    float f1=10.1,f2=20.1;
    g_swap(&a,&b,sizeof(a));
    g_swap(&c1,&c2,sizeof(c1));
    g swap(&f1,&f2,sizeof(f1));
    printf("a=%d b=%d\n",a,b);
     printf("c1=%d c2=%d\n",c1,c2);
     printf("f1=%f f2=%f\n",f1,f2);
void g swap(void*ptr1,void*ptr2,int check)
     void*cptr=malloc(check);
    memcpy(cptr,ptr1,check);
     memcpy(ptr1,ptr2,check);
    memcpy(ptr2,cptr,check);
```